

Polycopié de travaux dirigés

École doctorale GAO

Initiation aux statistiques avec le logiciel R

`christophe.ambroise@genopole.cnrs.fr`
`julien.chiquet@genopole.cnrs.fr`

Semestre de printemps 2012
Université d'Évry Val d'Essonne

Avant-propos

Fabricando fabri fimus.^a

a. la pratique fait l'ouvrier.

Outre les travaux dirigés, ce polycopié regroupe un ensemble de documents utiles à la pratique du logiciel R, en particulier dans l'optique d'analyses statistiques usuelles en biologie.

La première partie contient les énoncés des travaux dirigés à mener à bien au cours de cette formation : la première séance vous permettra de prendre vos marques sous R en découvrant la création d'objets et leur manipulation. Lors de la deuxième séance, nous explorons quelques unes des fonctions de haut niveau disponibles pour le traitement élémentaire des données (importation, sorties graphiques, fonctions de statistiques descriptives). Puis, les séances suivantes visent à découvrir les opérateurs de statiques permettant de mener à bien quelques uns des tests d'hypothèses les plus courants en biologie : nous pratiquerons notamment les tests du χ^2 , de Student, l'analyse de la variance et nous survolerons quelques unes des possibilités offertes par R pour le modèle linéaire. Pour vous aider à vous repérer, la difficulté des exercices en fonctions des connaissances acquises est indiquée à l'aide des symboles ☺, ☹, ☺.

La deuxième partie contient *une correction possible* des travaux dirigés. Il n'est pas *du tout* dans votre intérêt de consulter cette correction avant la fin d'une séance : votre formation en pâtirait¹. En cas de blocage, préférez toujours faire appel à l'enseignant ! Ces corrections doivent être vues comme une base d'exemples disponibles pour vos études R ultérieures, lorsque vous appliquerez ces enseignements à vos propres jeux de données.

Enfin, la troisième partie contient une série de documents liés à R et aux statistiques : des références, quelques rappels et les tables des lois usuelles. Une liste des commandes les plus courantes de R est également disponible à la page <http://stat.genopole.cnrs.fr/~jchiquet>

Remarque. Beaucoup d'exercices ont été adaptés des livres de Christophe Ambroise (G.F. McLachlan 2004) et Bernard Prum (Prum 1996).

Éléments bibliographiques

R est un logiciel bien documenté. Citons les guides officiels, indispensables à la bonne connaissance du langage (syntaxe et grammaire) :

- *An Introduction to R* (Venables et al. 1999–2009),
- *R language definition* (Team 1999–2009c),
- *R data Import/Export* (Team 1999–2009a).

1. Confer locution latine ci-dessus !

Pour une utilisation plus avancée, telle l'administration ou l'écriture de ses propres extensions, on citera

- *Writing R extension* (Team 1999–2009d),
- *R installation and administration* (Team 1999–2009b).

Il existe énormément de (bonnes) documentations « non-officielles », souvent dédiées à une utilisation particulière de R. Citons Paradis (2009), Verzani (2009) et l'ouvrage de référence² de W.N. Venables (2002).

Environnement de travail et conseils d'implémentation

Sous environnement LINUX, le terminal est un outil efficace de gestion des fichiers, des tâches et des programmes en cours d'exécution. Pour les réfractaires, le gestionnaire de fenêtre n'a rien envier à ceux des autres systèmes.

Créez un répertoire, par exemple³ R, à la racine de votre répertoire `home`. Placez-y des sous répertoires pour stocker vos données (`data`), vos scripts (`work`) et vos fonctions (`functions`).

Une fois à la racine de votre environnements de travail `/home/nom/R/work`, lancez R en tapant tout simplement la commande `R`. Ouvrez un autre onglet dans votre terminal dans le même répertoire pour pouvoir faire les manipulations usuelles de fichiers et lancer votre éditeur de texte préféré. *Surtout*,

- aidez-vous de la liste de commandes usuelles,
- abusez des commandes `help`, `?`, `help.search`, `apropos`, etc.,
- pensez à regarder les exemples de l'aide pour leur côté pratique,
- ayez sous les yeux un document pour vous aider (soit imprimé, soit ouvert dans un onglet de votre navigateur web).

Une fois familiarisé avec le logiciel, vous passerez à l'écriture de vos premiers véritables programmes. Lorsque la suite d'instructions requises pour le traitement d'un problème devient trop longue, il est nécessaire de la stocker dans un ou plusieurs fichiers pour ne pas avoir à tout réécrire dans le prompt. Utilisez un éditeur de votre choix pour écrire vos scripts et vos fonctions (Emacs est par exemple bien adapté et possède un mode spécifique à R appelé ESS).

Les fichiers externes (scripts ou fonctions) peuvent être chargés à l'aide de la commande `source`. Ainsi, `source("mes_fonctions.R")` charge toutes les fonctions contenus dans le fichier `mes_fonctions.R`, tandis que `source("mon_script.R")` exécute l'ensemble des commandes listées dans `mon_script.R`.

À vous de trouver l'équilibre qui vous convient entre l'interpréteur R, où la saisie est immédiatement évaluée, et la sauvegarde des commandes d'intérêt sous forme de scripts ou de fonctions.

Julien Chiquet,
31 janvier 2012

2. il s'agit cependant d'un ouvrage payant.

3. chacun a ses petites habitudes : si vous n'en avez pas je vous suggère les miennes.

Table des matières

Avant-propos	iii
Partie I : Travaux dirigés	1
1 Premiers pas sous R	3
2 Analyse statistique élémentaire	7
3 Introduction aux tests d'hypothèses sous R	11
4 Analyse de la variance sous R	15
5 Introduction au modèle linéaire sous R	17
Partie II : Corrections	21
1 Premiers pas sous R	23
2 Analyse statistique élémentaire	29
3 Introduction aux tests d'hypothèses sous R	43
4 Analyse de la variance sous R	53
5 Introduction au modèle linéaire sous R	63
Partie III : Documents	79
Références	81
A Tables statistiques	83
B Formulaire de statistique	93

Première partie

Travaux dirigés

Premiers pas sous R

À la découverte de la syntaxe et de l'esprit de programmation de R.

Exercice 1.1 (Génération de vecteurs ☺). Où l'on se familiarise avec la création de vecteurs (commandes `c()`, `seq()`, `rep()`, `paste()` et leurs options).

- i) Créer un vecteur contenant la suite des entiers de 1 à 12 de deux manières différentes.
- ii) Créer le vecteur `c(0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0)` de trois manières différentes.
- iii) Créer un vecteur contenant tous les multiples de 2 compris entre 1 et 50.
- iv) Créer un vecteur contenant tous les nombres de 1 à 100 qui ne sont pas des multiples de 5.
- v) Créer un vecteur contenant 3 fois chacun des 10 chiffres.
- vi) Créer un vecteur contenant une fois la lettre A, deux fois la lettre B, etc., 26 fois la lettre Z. Quelle est la longueur de cette suite ? (Utiliser la chaîne `LETTERS` prédéfinie).
- vii) Créer le vecteur `c("individu 1", "individu 2", ..., "individu 100")`.

Exercice 1.2 (Manipulation de séquences ☺). Où l'on se familiarise avec la manipulation de vecteurs (commandes `sample`, `length`, `sort`, `rev`, `sum`, `%in%`, `intersec`, `table`, etc.).

- Quels sont les entiers divisibles par 3 parmi les 100 premiers et combien y-en a-t-il ? Que vaut leur somme ? Leur produit ?
- Générer une séquence d'ADN de n bases. Compter le nombre d'occurrences de chaque lettre (d'abord sans puis avec la fonction `table`). Renvoyer les indices de la séquence où l'on trouve la lettre "t".¹
- Créer un vecteur contenant les 100 premiers entiers échantillonnés aléatoirement. À partir de ce vecteur, créer les vecteurs x et y des 100 premiers entiers ordonnés dans l'ordre croissant et décroissant. Concatenez x et y , enlever le seul nombre apparaissant deux fois de suite en le repérant à l'aide de la commande `diff`.
- On rappelle que

$$e^x = \sum_{k \geq 0} \frac{x^k}{k!}.$$

1. Pour aller plus loin, la bibliothèque `Biostrings` s'impose !

Créer dans un vecteur `exp2` les 20 premiers termes de cette suite. Supprimer toutes les valeurs inférieures à 10^{-8} . En déduire une approximation de e^2 et comparer avec la valeur `exp(2)`.

- Créer un vecteur `couleurs` contenant 10 couleurs de votre choix sous forme de chaînes de caractères. Créer un échantillon aléatoire de taille 4 parmi ces couleurs ainsi qu'un vecteur `primaires` contenant les 3 couleurs primaires. Tester combien et quelles sont les couleurs primaires présentes dans votre échantillon.

Exercice 1.3 (Jeux de hasard ☺).

1. On veut mimer un jeu de pile ou face. On notera 1 pour pile et 0 pour face. Simulez 1000 lancers de pièces. On gagne 1 € si c'est pile et on perd 1 € si c'est face. Combien d'argent avez-vous gagné à l'issue des 1000 lancers ?
2. Un ami vous propose le jeu suivant. On lance un dé. Si le résultat est 5 ou 6, on gagne 3 €, si le résultat est 4 on gagne 1 € et si c'est 3 ou moins on perd 2.5 €. Avant d'accepter la partie, vous essayez de simuler ce jeu, pour voir si vous avez des chances de vous enrichir. Conclusion ?

Exercice 1.4 (Tableaux de données ☺). On suppose que la taille et le poids des individus en France se répartissent selon des lois normales de paramètres suivants :

- la taille des femmes est en moyenne de 165 cm, avec un écart-type de 6 cm.
- la taille des hommes est en moyenne de 175 cm, avec un écart-type de 7 cm.
- le poids des femmes est en moyenne de 60 kg, avec un écart-type de 2 kg.
- le poids des hommes est en moyenne de 75 kg, avec un écart-type de 4 kg.

Créer les vecteurs `taille.homme`, `taille.femme`, `poids.homme`, `poids.femme` contenant la taille et le poids de $n = 257$ hommes et $m = 312$ femmes générés selon les lois ci-dessus (on supposera que poids et taille sont deux variables indépendantes ce qui est bien entendu faux!).

Créer un tableau `donnees` à $n + m$ lignes et 3 colonnes telles que

- la première colonne contienne la variable taille,
- la deuxième colonne contienne la variable poids,
- la troisième colonne soit un facteur indiquant le sexe de l'individu.

Placer l'objet `donnees` dans l'itinéraire de recherche à l'aide de la commande `attach`. Puis, à l'aide de la commande `by`,

- déterminer simultanément le plus petit poids chez les femmes et le plus petit poids chez les hommes, ainsi que le numéro des individus correspondant,
- de même pour la taille,
- faites un résumé statistique de chacun des deux groupes (commande `summary`).

Y a-t-il des hommes de plus d'un mètre quatre vingt dix et de moins de soixante-quinze kilos ? Si oui, combien ? Y a-t-il des femmes de moins d'un mètre soixante et de plus de soixante kilos ? Combien ?

Exercice 1.5 (Population de bactéries ☺). On souhaite modéliser la croissance d'une population bactérienne mise en culture dans une boîte de Petri. À cet effet, on distingue deux types de bactérie :

1. des bactéries jeunes et « immatures », notées a , qui ne se divisent pas ;
2. des bactéries « matures », notées b , susceptibles de se diviser par mitose.

On suppose que la reproduction a lieu à intervalles de temps discrets ; les bactéries b se divisent d'un instant à l'autre en une bactérie a et une bactérie b ; enfin, toute bactérie a devient mature d'un pas de temps à l'autre.

Questions

1. On note $n_a(t)$ et $n_b(t)$ le nombre de bactéries de chaque type à l'instant t . Écrire le système de deux équations décrivant l'évolution de $n_a(t + 1)$ et $n_b(t + 1)$ en fonction de $n_a(t)$ et $n_b(t)$.
2. Écrire une fonction `PopBacteries(n0,T)` qui renvoie trois vecteurs de taille $T + 1$ contenant l'évolution des deux catégories de bactérie de l'instant initial au temps T ainsi que l'évolution de la population totale. Le paramètre n_0 est le nombre $n_a(0)$, et l'on suppose que $n_b(0) = 0$.
3. Pour $T = 20$ et $n_0 = 1$, générer la population bactérienne correspondante et calculer le taux d'accroissement de la population totale. Représenter graphiquement ces résultats (fonction `plot`).
4. On souhaite maintenant introduire de l'aléa dans la dynamique bactérienne. À cet effet, on suppose qu'une bactérie de type b a une probabilité p d'accomplir une mitose en $a + b$. Modifier la fonction `PopBacteries(n0,T,p)` en ajoutant le paramètre p .
5. Étudier l'évolution de la population et son taux de croissance totale pour diverses valeurs de p .

Analyse statistique élémentaire

Nous abordons la manipulation pratique de tableaux de données et les outils de statistiques descriptives de R. Nous étudions le cas de données catégorielles et/ou numériques. Ceci est l'occasion de faire quelques rappels de statistiques.

Exercice 2.1 (Bières ☺). Un sondage est réalisé auprès de 100 individus pour savoir où va leur préférence parmi un panel représentatifs de marques de bière. Les résultats obtenus se trouvent dans le fichier `bieres.csv`.

1. Lire le fichier de données sous forme de `data.frame`.
2. Combien de marques sont considérées ? Quelles sont-elles ?
3. Compter les occurrences de chacune des marques de bières. Les représenter sous la forme de graphe en barres. Représenter cette distribution sous forme de camembert en choisissant les couleurs vous même. Utiliser une seule fenêtre graphique pour les deux figures.

Commandes utiles : `levels`, `nlevels`, `table`, `barplot`, `pie`, `par`.

Exercice 2.2 (Somnifère ☺). Pour étudier l'effet d'un somnifère, on mesure chez 20 patients le nombre d'heures de sommeil supplémentaires par rapport à la durée moyenne de leur nuit sans traitement. On obtient les résultats suivants :

# patient	1	2	3	4	5	6	7	8	9	10
extra	0.7	-1.6	-0.2	-1.2	-0.1	3.4	3.7	0.8	0.0	2.0
# patient	11	12	13	14	15	16	17	18	19	20
extra	1.9	0.8	1.1	0.1	-0.1	4.4	5.5	1.6	4.6	3.4

TABLE 2.1 – données somnifères

1. Saisir ces données dans un vecteur `x`.
2. Calculer la moyenne empirique \bar{x} , la variance empirique s^2 , la variance empirique corrigée s^{*2} et l'écart-type, i.e.,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2, \quad s^{*2} = \frac{n}{n-1} s^2,$$

uniquement avec les commandes `sum` et `length`. Comparer vos résultats avec les commandes `mean`, `var` et `sd` (que vous pourrez directement utiliser dorénavant).

3. On définit la médiane d'un échantillon par tout réel le séparant en deux parties égales. Calculer une valeur possible de la médiane, et observer la valeur renvoyée par la fonction de R.
4. Tracer un diagramme en tige et feuille.
5. Tracer la fonction de répartition empirique puis l'histogramme normalisé des données dans la même fenêtre graphique.
6. Ces données sont en fait issues de deux groupes d'individus : apposer une variable indiquant le groupe associé à l'observation de la variable `extra` sachant que les 10 premiers individus sont issus du groupe 1 et les 10 suivants du groupe 2 (utiliser, par exemple, la commande `data.frame`). Faire un résumé statistique pour chaque groupe et tracer alors les boîtes à moustaches des observations selon les groupes. Qu'en pensez-vous ? Comment conclure correctement ?

Commandes utiles : scan, median, stem, par, ecdf, hist, summary, boxplot.

Exercice 2.3 (iris – description multidimensionnelle ☺). Ce jeu de données, initialement étudié par Fisher, est composé de 150 iris provenant de trois famille de fleur différentes (Virginia, Versicolor et Setosa), sur lesquelles ont été mesurées les longueurs et les largeurs du sépale et du pétale.

1. Sous R, charger ces données à l'aide de la commande `data(iris)`. La commande `head(iris)` permet d'afficher les 6 premières lignes du tableau : combien y a-t-il DE variables quantitatives ? Catégorielles ? Quelles sont-elles ? On note `x` le tableau individus/variables associé aux variables quantitatives. Créer l'objet R correspondant. Créer une variables catégorielle `espece`.
2. Faites un résumé numérique des colonnes de `x`. À l'aide de la commande `by`, faites un résumé de chaque espèce de fleur.
3. Faites une boîte à moustache pour chaque variable de chaque espèce (4 graphes de 3 boîtes à moustaches).
4. Tracer les graphes paire à paire de la matrice `x` à l'aide de la commande `pairs`
5. Calculer les matrices de variance-covariance et de corrélation. Quel commentaires pouvez-vous faire en regard des graphes de la question précédente ?

Commandes utiles : summary, head, by, boxplot, cov, cor.

Exercice 2.4 (Puces à ADN ☺). Les distributions des intensités moyennes des spots d'ADNc correspondants aux gènes exprimés et non exprimés peuvent être modélisées par deux gaussiennes. On suppose que la première distribution a une espérance $\mu^e = 1000$ et un écart type $\sigma^e = 100$; la seconde distribution a pour espérance $\mu^{ne} = 400$ et pour écart type $\sigma^{ne} = 150$.

Chaque gène correspond à 4 spots répliqués. L'expression d'un gène est définie comme la moyenne des 4 spots qui lui sont associés.

1. Analyse élémentaire du modèle
 - (a) Créer sous R les variables `mu.e`, `sigma.e`, `mu.ne` et `sigma.ne` et affectez-y les valeurs de l'énoncé.
 - (b) On note S^e la variable aléatoire décrivant l'intensité d'un spot correspondant à un gène exprimé. Quelle est la probabilité pour que S^e ait une valeur inférieure ou égale à 700 ?

- (c) On note G^e la variable aléatoire décrivant le niveau d'expression d'un gène exprimé. Quelle est la probabilité pour que G^e ait une expression inférieure ou égale à 700 ?
 - (d) On introduit la variable aléatoire G^{ne} pour les gènes non exprimés. Quelle est la valeur seuil t telle que la probabilité d'avoir G^e inférieure ou égale à t soit égale à la probabilité d'avoir G^{ne} supérieure à t ?
 - (e) Quelle est la probabilité d'avoir un gène exprimé dont l'expression est inférieure à t (faux négatif) ?
 - (f) Quelle est la probabilité d'avoir un gène non exprimé dont l'expression est supérieure à t (faux positif) ?
2. Simulations et graphiques
- (a) Générer $n = 1000$ intensités de spots correspondant aux gènes exprimés et non exprimés. Les stocker dans les vecteurs `spots.e` et `spots.ne`. Parmi tous les spots générés, stocker la plus petite et la plus grande valeur observée dans des variables `MIN` et `MAX`.
 - (b) Créer deux objets de classe histogramme, sans les tracer, correspondant à chacune des deux populations de spots et stocker les dans des variables `hist.e` et `hist.ne`.
 - (c) Tracer sur un même graphique les deux histogrammes normalisés et les densités théoriques (fonction `curve`). Utiliser deux couleurs différentes pour les deux populations de spots. Apposer une légende au graphe (commande `legend`).
 - (d) Tracer sur un même graphique les densités théoriques des gènes exprimés et non exprimés. Faire une légende. Puis, à l'aide de la commande `polygon`, représenter l'aire sous courbe correspondant à la probabilité pour qu'un gène non exprimé ait une expression inférieure à 300. Enfin, tracer une droite verticale indiquant l'emplacement du seuil t (commande `abline`).

Exercice 2.5 (Maximisation numérique de la vraisemblance ☺☺). En statistique, une manière usuelle de déterminer l'estimateur d'un paramètre d'une distribution de probabilité consiste à chercher la valeur de ce paramètre qui maximise la *fonction de vraisemblance* : si X est une variable aléatoire à densité de probabilité $f(x; \theta)$ où θ sont les paramètres de la fonction f , et que l'on observe x_1, x_2, \dots, x_n valeurs de X , alors la fonction de vraisemblance est définie par

$$L(\theta; x_1, \dots, x_n) = \prod_{i=1}^n f(x_i; \theta).$$

L'idée consistant à utiliser comme estimateur la valeur de θ qui maximise la fonction L est naturelle : si on fait confiance en notre modèle $X \sim f(\cdot; \theta)$ pour décrire le phénomène d'intérêt, on cherche une valeur de θ qui rende les $f(x_i; \theta)$ le plus élevé possible, c'est-à-dire les plus *vraisemblables* par rapport aux observations. La valeur de θ pour laquelle L est maximale est appelée *estimateur du maximum de vraisemblance*. Souvent, on l'obtient en maximisant de manière équivalente la log-vraisemblance puisque l'on préfère les additions aux multiplications :

$$\log L(\theta; x_1, \dots, x_n) = \sum_{i=1}^n \log f(x_i; \theta).$$

Cette fonction mesure le niveau d'adéquation du modèle aux données en fonction de la valeur des paramètres θ du modèle.

En biologie et dans tous les champs d'application de la statistique, il est courant que le modèle $X \sim f(\cdot; \theta)$ choisi pour le phénomène d'intérêt soit trop complexe pour permettre le calcul et la maximisation analytique de L ou $\log L$. On doit donc avoir recours à des techniques *numériques*. Nous allons étudier un exemple de modèle où la maximisation est soluble analytiquement, ce qui nous permettra de vérifier que la résolution numérique fonctionne.

Considérons un phénomène X (par exemple, la concentration de protéine en présence dans une solution) modélisé par une loi normale $\mathcal{N}(\mu, \sigma^2)$. La fonction de densité est donnée par

$$f(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x - \mu)^2}{2\sigma^2}\right\}.$$

Questions préliminaires¹

1. Calculer analytiquement $\log L(x_1, \dots, x_n; \mu, \sigma^2)$.
2. Déterminer les estimateurs du maximum de vraisemblance en dérivant successivement $\log L$ par rapport à μ et σ^2 . Que pensez-vous du résultat ?

Maximisation numérique

1. Générer un jeu de données gaussien de taille $n = 100$, de moyenne $\mu = \pi/2$ et d'écart type $\sigma = \sqrt{2}$. Calculer les valeurs prises par les estimateurs du maximum de vraisemblance de μ et σ^2 obtenus dans les questions précédentes.
2. Écrire une fonction `loglikelihood` qui prend en argument `x`, `mu`, `sigma` et renvoie la valeur de la fonction de log-vraisemblance pour (x_1, \dots, x_n) , μ et σ donnés.
3. À l'aide de la fonction `optimize`, déterminer numériquement les valeurs de μ et σ maximisant la fonction `loglikelihood`.
4. Dans une même fenêtre graphique, représenter
 - l'histogramme des données,
 - la fonction $\log L$ pour σ fixée à sa vraie valeur, en faisant varier μ sur $[\pi/2 - \varepsilon, \pi/2 + \varepsilon]$; situer également les valeurs estimées analytiquement et par `optimize` à l'aide de `abline`.
 - la fonction $\log L$ pour μ fixée, en faisant varier σ sur $[\sqrt{2} - \varepsilon, \sqrt{2} + \varepsilon]$; de même, situer les valeurs estimées analytiquement et par `optimize`. Représenter également la variance empirique corrigée.
5. Créer une matrice `logL` contenant les valeurs de la log-vraisemblance en faisant varier à la fois μ et σ . Créer une liste `data=list(x,y,z)` et utiliser la fonction `my2dplot(data)` (fournie par le prof!) pour représenter la vraisemblance en 3D.

Ouf!

1. s'aider du prof en cas de galère...

Introduction aux tests d'hypothèses sous R

Ce TD aborde les tests du χ^2 d'adéquation et d'indépendance, ainsi que le test de Student appliqué au test de la valeur du paramètre d'espérance et de l'égalité des espérances entre deux populations. Le test de Fisher d'égalité des variances est également étudié.

Exercice 3.1 (H1N1 ☺). Un laboratoire d'épidémiologie a publié une étude relative à l'âge des individus touchés par la grippe A lors de la première vague, qui s'est terminée fin décembre 2009. La répartition suivante a été observée :

âge	de 0 à 5	de 5 à 18	de 18 à 35	de 35 à 65	65 et plus
%age	7.5%	12.5%	35%	30%	15%

Au début de la deuxième vague (janvier 2010), les services municipaux de la ville d'Évry participent à une nouvelle étude : si la répartition de l'âge des individus touchés est significativement différente de celle obtenue lors de la première vague, ceci peut indiquer une mutation du virus. La réponse des services hospitaliers devra être adaptée le cas échéant.

Du 1^{er} au 15 janvier, 237 personnes ont été touchées à Évry. La classe d'âge de chaque individu est répertoriée dans le fichier `grippe.csv`.

D'après ces observations, les individus sont-ils touchés de la même manière que lors de la première vague ? Répondez par un test du χ^2 d'adéquation, d'abord « à la main », puis à l'aide de la fonction `chisq.test`.

Exercice 3.2 (Alcool au volant ☺). En cas d'accident de la route, le taux d'alcoolémie est systématiquement relevé chez les blessés. Dans ce contexte, la sécurité routière mène une enquête auprès de 975 accidentés de la route et communique les résultats de son enquête (fichier `accidents.rda`).

Montrer, par un test du χ^2 d'indépendance, que le taux d'alcoolémie influe sur la gravité des blessures.

Exercice 3.3 (pharmacologie ☺). On désire expérimenter l'action d'un produit pharmaceutique sur des souris. Le temps de réaction μ des souris à un test déterminé est supposé être une variable aléatoire de loi gaussienne, le temps de réaction moyen étant de 19 minutes dans des conditions normales. On administre à dix souris une dose du produit pharmaceutique et on leur fait passer le test. On obtient les temps de réaction suivants : 15 mn, 14 mn, 21 mn, 12 mn, 17 mn, 12 mn, 19 mn, 18 mn, 20 mn, 21 mn.

Remarque. Dans ce qui suit, vous pourrez utiliser la fonction `t.test` de R pour vérifier vos résultats. La fonction `qt` permet le calcul des fractiles d'une loi de Student et `pt` le calcul de sa fonction de répartition.

Utiliser la fonction `assocplot` pour détailler la nature des dépendances entre ces deux variables

1. Saisir les données sous la forme d'un vecteur `x`. Calculer \bar{x} , s et s^* .
2. Écrire une fonction `int.conf.mu` qui revoie une liste contenant l'estimateur $\hat{\mu}$, la borne inférieure `inf` et la borne supérieure `sup` de l'intervalle de confiance. Elle prend en arguments
 - un vecteur de données `x`,
 - un scalaire `niveau` pour le coefficient de sécurité de l'intervalle,
 - une chaîne de caractères `alternative` pouvant prendre les valeurs "two.sided", "less" ou "greater" selon la forme désirée.
3. Utiliser cette fonction pour donner des intervalles de confiance avec un coefficient de sécurité de 90%, 95% et 99% du temps moyen de réaction des souris pour l'alternative bilatère.
4. On s'attend à ce que le produit ait un effet stimulant et donc que le temps de réaction des souris ayant reçu une dose soit en moyenne inférieur à celui des souris n'en ayant pas reçu. Tester au niveau 5% l'effet du produit à l'aide de `t.test`. Calculer la valeur du seuil au niveau choisi ainsi que la p -valeur. Proposer un intervalle de confiance unilatère pour le paramètre μ .
5. Calculer la valeur de la puissance π pour diverses valeurs de μ_1 , le paramètre sous l'hypothèse alternative. Tracer la courbe de puissance. Faire apparaître le seuil de rejet de la question précédente.

Exercice 3.4 (pipette ☺). On a prélevé une solution plusieurs fois en utilisant deux pipettes calibrées de même volume. On a pesé le contenu du volume délivré par la pipette. Les résultats des différents pipettages, exprimés en grammes, sont les suivants :

Pipette 1	0.0987	0.0990	0.0996	0.0995	0.0998	0.0984
Pipette 2	0.1016	0.1008	0.1002	0.0995	0.0990	0.1023

1. Créer 2 vecteurs `pip1` et `pip2`. Tracer les boîtes à moustaches pour chacune des deux pipettes. Qu'en conclure ?
2. Les deux pipettes ont-elles la même précision de mesure ?
3. Les quantités prélevées sont-elles comparables ?
4. Vérifier l'équivalence avec une anova à un facteur (fonction `anova` ou `aov`).

Exercice 3.5 (test de Kolmogorov-Smirnov). Un fabricant garantit que la fiabilité des appareils qu'il vend est telle que leur durée de vie suit une loi exponentielle, de moyenne 1700 h. Afin de tester cette affirmation, la durée de vie en heures de 10 de ces appareils pris au hasard a été mesurée. Les durées de vie observées sur cet échantillon de 10 appareils sont les suivantes

1. (a) Tracer la fonction de répartition empirique associée à l'échantillon des $n = 10$ appareils.

160 330 600 780 1030 1350 1950 2400 3400 5000

TABLE 3.1 – durées de vie

- (b) Tracer sur le même graphique la fonction de répartition de la loi exponentielle $\mathcal{E}(1/1700)$.
2. Test de Kolmogorov-Smirnov
- (a) Calculer pour chaque valeur observée t_i , les écarts entre les fréquences cumulées expérimentales $\mathbb{P}(T < t_i)$ et $\mathbb{P}(T \leq t_i)$ et les fréquences cumulées théoriques pour la loi exponentielle de paramètre 1/1700. La valeur de $D_n = \sup_{t \in \mathbb{R}} |\hat{F}_n(t) - F_0(t)|$ est le maximum de ces écarts (en valeur absolue). Quelle est la valeur observée de cette statistique ?
- (b) À partir de ces données, pensez-vous que le fabricant puisse être accusé de faire de la publicité mensongère ?

3.1 Exercices issus de l'ouvrage « Using R for introductory statistics »

Exercice 3.6. Le jeu de données `vacation` décrit le nombre de jours de congés payés pris par des ouvriers dans une industrie textile. Utiliser le test approprié permettant de déterminer si le nombre légal de jour de congé payé (24) est globalement respecté.

Exercice 3.7. Le jeu de donnée `smokyph` décrit le niveau de pH d'échantillon d'eaux relevés dans les Alpes. Utiliser la colonne `smokyph$waterph` pour tester l'hypothèse selon laquelle le pH est neutre.

Exercice 3.8. Un sondage d'opinion réalisé auprès de 900 personnes représentative de la population française indique 440 votant en faveur de Nicolas Sarkozy en cas d'affrontement avec Francois Hollande lors du second tour des élections présidentielle. Est-ce que ces données soutiennent l'hypothèse selon laquelle Nicolas Sarkozy a reçu 50% des votes ?

Exercice 3.9. Le jeu de données `cancer` (colonne `stomach`) décrit les temps de survie de patient atteint de cancer de l'estomac recevant de grande doses de vitamines C. Tester l'hypothèse nulle selon laquelle la médiane des temps de survie est 100. Un t-test est-il également envisageable ?

Exercice 3.10. Le jeu de données `homework` mesure le temps de travail hebdomadaire d'étudiants issus d'écoles publiques et privés. Utiliser le test adéquate pour déterminer s'il existe une différence significative entre les centres de distribution.

Exercice 3.11. Charger les données `corn` : celles-ci décrivent les rendements de 12 champs de maïs, tout d'abord pour l'espèce classique puis pour une nouvelle graine transgénique. Utiliser le test le plus approprié pour déterminer si la nouvelle graine induit un meilleur rendement.

Exercice 3.12. Les instituts de sondage souhaite évaluer l'impact des annonces de Nicolas S. en réalisant à nouveau un sondage d'opinion auprès de 900 personnes représentatives : 475 se déclarent alors favorables : Conclusions ?

Analyse de la variance sous R

Ce TD propose deux exercices type analyse de la variance, à un et deux facteurs explicatifs.

Exercice 4.1 (Asthme ☹). On souhaite comparer trois traitements notés A, B, C contre l'asthme : le traitement B est un nouveau traitement, que l'on met en compétition avec les traitements classiques A et C. On répartit par tirage au sort les patients et on mesure sur chacun la durée en jours avant la prochaine crise d'asthme.

1. Visualisation des données.
 - (a) Stocker les données `asthme.dat` dans une variable de votre choix à l'aide de la fonction `read.table`. La table ainsi créée a deux colonnes : l'une contenant le délai observé avant la prochaine crise d'asthme, l'autre le type de traitement reçu.
 - (b) Faire un résumé numérique de l'ensemble des données puis par traitement. Représenter graphiquement ces résultats à l'aide de boîtes à moustaches. Que peut-on en conclure ?
2. Analyse de la variance.
 - (a) Créer une fonction `somme.carres` qui prend en argument un vecteur d'observations et un vecteur de facteurs. La fonction renvoie une liste contenant
 - la somme des carrés totale et le nombre de degrés de liberté associé,
 - la somme des carrés résiduelle et le nombre de degrés de liberté associé,
 - la somme des carrés des facteurs et le nombre de degrés de liberté associé.
 - (b) Calculer la valeur observée de la statistique de test de l'anova 1, la valeur du seuil de rejet au niveau α de votre choix et la valeur de la p -valeur.
 - (c) Comparer les résultats avec ceux de la fonction `anova` de R.
3. Étude de contrastes.
 - (a) On note μ_A , μ_B et μ_C les délais moyens obtenus avec les traitements A, B et C. Tester l'hypothèse selon laquelle $\mu_A = \mu_C$, puis $\mu_A = \mu_B$. On utilisera comme estimateur de la variance la valeur des CMR obtenu à la question précédente. Conclure.
 - (b) On souhaite déterminer l'apport du traitement B par rapport au traitement A. À cet effet, nous allons étudier le contraste C_{AB} . Construire un intervalle bilatère puis unilatère à 95% pour C_{AB} . Conclusion ?

Exercice 4.2 (Rendement de blé ☺). On mesure les rendements en blé sur 21 parcelles de même aire représentant 6 niveaux :

niveau	1	2	3	4	5	6
	53.4	76.9	55.1	71.6	90.1	76.4
	64	62.4	72	80.4	89.6	77
	68.3	70.8	67	67.4	81.6	77.8
	65.5	70			82.2	

TABLE 4.1 – données blé

- On note $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$ les espérances de rendements pour les divers niveaux. On considère le modèle $H_6 : \mu_1 \neq \mu_2 \neq \mu_3 \neq \mu_4 \neq \mu_5 \neq \mu_6$.
 - Créer un vecteur `donnees` contenant les observations et un vecteur `niveaux` contenant les facteurs associés au modèle H_6 .
 - Représenter les boîtes à moustaches et les dotplot pour les différents groupes à l'aide des fonctions `boxplot` et `dotchart`.
 - Tester $H_1 : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6$ contre H_6 à l'aide d'une anova 1 qu'on fera « manuellement » (calculs de SCT , SCR , SCE , du seuil à 5% et 1% ainsi que la p -valeur).
 - Comparer les résultats avec la fonction `aov` de R.
- Les niveaux 1,2,3 correspondent à une région «Nord», les niveaux 4,5,6 à une région «Sud». Même question qu'en 1 en testant H_1 contre $H_2 : \mu_{\text{nord}} \neq \mu_{\text{sud}}$, en utilisant directement la fonction `aov`. Le vecteur des facteurs est noté `regions`.
- Les niveaux 1 et 4 correspondent à une variété A d'espérance μ_A , les niveaux 2 et 5 correspondent à une variété B d'espérance μ_B et les niveaux 3 et 6 correspondent à une variété C d'espérance μ_C . Même question qu'en 1 en testant H_1 contre $H_3 : \mu_A \neq \mu_B \neq \mu_C$, en utilisant directement la fonction `aov`. Le vecteur des facteurs est noté `varietes`.
- Refaire le problème en anova 2 avec comme facteurs «régions» et «variétés» en utilisant la fonction `aov`. Utiliser la fonction `interaction.plot` pour faire votre choix entre le modèle avec ou sans interactions.

Introduction au modèle linéaire

SOUS R

Au cours de ce TD, nous survolons quelques unes des fonctionnalités offertes par R pour le modèle linéaire.

Exercice 5.1 (Performances ☺). On s'intéresse aux performances sportives d'enfants de 12 ans. Chaque enfant passe une dizaine d'épreuves (courses, sauts, lancers, etc.), et les résultats sont synthétisés dans un indice global, noté Y . On cherche à mesurer l'incidence sur ces performances de deux variables : la capacité thoracique X_1 et la force musculaire X_2 . Ces trois quantités, Y , X_1 et X_2 , sont repérées par rapport à une valeur de référence, notée à chaque fois 0, les valeurs positives étant associées aux « bonnes » performances.

Les mesures associées à un échantillon de 60 enfants sont stockées dans le vecteur `data`, dont vous disposerez sous R une fois chargé le fichier `perf.dat`.

On adopte le modèle à deux paramètres

$$H_2 : Y = a_1 X_1 + a_2 X_2 + \varepsilon,$$

où ε est un résidu non expliqué : les ε_i associés aux différents individus seront modélisés par des $\mathcal{N}(0, \sigma^2)$ indépendantes (Notons que le « calage » des données autour de zéro se traduit par le fait que, quand $X_1 = X_2 = 0$, alors $\mathbb{E}(Y) = 0$).

1. Représenter le nuage de points et les graphes comparés des données à l'aide des fonctions `pairs` et `cloud` de la bibliothèque `lattice`.
2. Estimer les paramètres a_1 et a_2 du modèle, en écrivant les équations aux paramètres sous forme matricielle. Résoudre le système associé à l'aide de la commande `solve`. Vérifiez vos résultats avec la fonction `lm` de R.
3. À l'aide de la bibliothèque `scatterplot3d`, représenter le nuage de point et le plan estimé par le modèle.
4. Tester H_2 contre H_0 : conclusion ?
5. On adopte maintenant le modèle

$$H_1 : Y = a X_1 + \varepsilon.$$

Estimer a et représenter les données et la droite de régression associée. Enfin, vous testerez H_1 contre H_0 .

Exercice 5.2 (Le sida du chat ☹). On mesure le taux de leucocytes T_4 chez le chat X jours après avoir inoculé à l’animal le virus FeLV, analogue du HIV. On appelle Y le logarithme de ce taux. Le tableau suivant donne les mesures faites sur $n_1 = 17$ chats mâles et $n_2 = 15$ chattes. Les données de la table 5.1 sont disponibles dans le fichier `chat.dat`

1. On définit le modèle H_4 comme celui où, pour chaque sexe, Y varie linéairement en fonction de X :
 - pour les mâles, $Y = a_1X + b_1 + \varepsilon$
 - pour les femelles, $Y = a_2X + b_2 + \varepsilon$.

Pour chaque groupe, ajuster la droite de régression. Tester l’égalité des variances des résidus.

2. On définit le modèle H_2 comme celui où une droite de régression commune explique les mesures des deux sexes : $Y = aX + b + \varepsilon$. Tester H_2 contre H_4 .
3. Comparer séparément le modèle retenu aux modèles suivants :
 - H_a : « $a_1 = a_2$ », b_1 et b_2 quelconques, i.e.,

$$Y_i = a X + b_1 \mathbb{1}_{\{\text{mâle}\}} + b_2 \mathbb{1}_{\{\text{femelle}\}} + \varepsilon_i.$$

- H_b : « $b_1 = b_2$ », a_1 et a_2 quelconques, i.e.,

$$Y_i = a_1 \mathbb{1}_{\{\text{mâle}\}} X + a_2 \mathbb{1}_{\{\text{femelle}\}} X + b + \varepsilon_i.$$

	mâles		femelles		
	X	Y	X	Y	
	44	4.66	84	3.45	
	317	3.08	47	3.89	
	292	1.28	20	3.79	
	179	3.17	209	3.79	
	39	5.59	106	3.81	
	257	2.88	343	0.61	
	354	1.60	325	2.04	
	349	3.48	346	0.41	
	195	3.39	151	2.67	
	245	3.47	267	0.89	
	270	3.20	80	4.39	
	166	2.90	249	2.56	
	57	4.83	341	0.28	
	198	2.96	189	2.43	
	20	5.17	50	3.85	
	187	3.44			
	270	3.18			
n	17		15		32
$\sum X_i$	3 439		2 807		6 246
$\sum X_i^2$	883 645		724 565		1 608 210
$\sum Y_i$	58.28		38.86		97.14
$\sum Y_i^2$	220.0506		129.5332		349.5838
$\sum X_i Y_i$	10184.95		5135.18		15 320.13

TABLE 5.1 – données chat

Exercice 5.3. Présidentielles ☹

Le jeu de données `gavote` décrit le vote présidentiel aux états-unis en 2000, dans l'état de Géorgie. Chacun des 159 "canton" est décrit par les variables suivantes :

- `equip` Le système physique de vote
 - `LEVER` : machine à levier
 - `OS-CC` : Scan optique comptage centralisé ("central count"),
 - `OS-PC` : Scan optique comptage local ("precinct count")
 - `PAPER` : vote par bulletin papier
 - `PUNCH` : vote par poinçon
- `econ` le statut économique du "canton" (`middle`, `poor`, `rich`).
- `perAA`, le pourcentage d'afro-américains
- `rural` indicateur de la ruralité du canton (`urban`, `rural`)
- `atlanta` indicateur de l'appartenance ou non à Atlanta
- `gore` nombre de votes pour Gore
- `bush` nombre de votes pour Bush
- `other` number of votes for other candidates
- `votes` nombre de votes validés
- `ballots` nombre de bulletins

1. Charger le jeu de données `gavote` et faites un résumé numérique des données.
2. Créer la variable `undercount` qui est la proportion de bulletins de vote considérés comme nuls. Représenter la distribution de cette nouvelle variable. Créer la variable `pergore` (pourcentage de votants pour Gore) et tracer le diagramme de dispersion croisant `pergore` avec `perAA`.
3. Tracer la droite de régression.
4. Représenter la distribution de la variable `equip`
5. Régresser `undercount` sur `perAA`. Interprétez ce modèle `model1`.
6. Calculer la somme des résidus au carré, le R^2 et le R^2 ajusté.
7. Régresser `undercount` sur `rural`. Interprétez ce modèle `model2`.
8. Centrer les variables `pergore` et `perAA`, et ajuster un modèle (`model3`) de régression linéaire qui explique `undercount` en fonction de `cperAA`, `cpergore`, `rural` et `equip`. Commentez.
9. Supprimer toutes les variables non significatives et comparer le modèle obtenu avec le précédent. Commentez.
10. En utilisant la procédure `step`, simplifier le modèle `model3`. Commentez.
11. Faites le diagnostique de la régression finale.

Deuxième partie

Corrections

Premiers pas sous R

Exercice 1.1 (Génération de vecteur).

1.

```
> 1:12
> seq(from = 1, to = 12, by = 1)
```
2.

```
> seq(from = 0.5, to = 5, len = 10)
> seq(from = 0.5, to = 5, by = 0.5)
> 1:10/2
```
3. L'opérateur %% correspond au reste de la division entière.

```
> v <- 1:50
> v[v%%2 == 0]
```
4.

```
> v <- 1:100
> v[v%%5 != 0]
```
5.

```
> rep(c(1:10), each = 3)
```
6.

```
> rep(LETTERS, c(1:length(LETTERS)))

> paste(rep("individu", 100), 1:100)
```

Exercice 1.2 (Manipulation de séquences).

1.

```
> integers <- 1:100
> div.by.3 <- integers[integers %% 3 == 0]
> cat("Nombre:", length(div.by.3), "somme:", sum(div.by.3),
+     "produit:", prod(div.by.3))

Nombre: 33 somme: 1683 produit: 4.827109e+52
```
2. On peut utiliser l'opérateur `sum` sur un vecteur de booléens :

```
> adn <- sample(c("a", "c", "g", "t"), 50, rep = TRUE)
> nbA <- sum(adn == "a")
> nbC <- sum(adn == "c")
> nbG <- sum(adn == "g")
> nbT <- sum(adn == "t")
> cat(nbA, "A,", nbC, "C,", nbG, "G,", nbT, "T.")

15 A, 12 C, 13 G, 10 T.
```

Cependant, la fonction `table` fait directement le travail :

```

> table(adn)

adn
 a c g t
15 12 13 10

> which(adn == "t")

[1]  3 12 15 16 21 22 23 28 34 36

3. > v <- sample(1:100)
   > x <- sort(v)
   > y <- rev(x)
   > v <- c(x, y)
   > v <- v[-which(diff(x) == 0)]

4. > exp2 <- 2^(0:20)/factorial(0:20)
   > exp2 <- exp2[exp2 >= 1e-08]
   > sum(exp2)

[1] 7.389056

> exp(2)

[1] 7.389056

5. > couleurs <- c("rouge", "bleu", "vert", "marron", "rose", "jaune",
+ "orange", "violet", "beige", "prune")
   > echantillon <- sample(couleurs, 4)
   > primaires <- c("rouge", "jaune", "bleu")
   > sum(primaires %in% echantillon)

[1] 1

> intersect(primaires, echantillon)

[1] "jaune"

```

Exercice 1.3 (Jeux de hasard).

```

1. > tirages <- sample(c("pile", "face"), 1000, rep = TRUE)
   > gains <- sum(tirages == "pile") - sum(tirages == "face")
   > gains

[1] 0

2. > tirages <- sample(c(1:6), 1e+05, rep = TRUE)
   > esp.emp <- 3 * sum(tirages >= 5) + sum(tirages == 4) - 2.5 *
+ sum(tirages <= 3)
   > esp.theo <- (2/6 * 3 + 1/6 * 1 - 3/6 * 2.5) * 1e+05
   > cat("Espérance théorique:", esp.theo, "estimée:", esp.emp)

Espérance théorique: -8333.333 estimée: -7362

```

Exercice 1.4 (Tableaux de données). Commençons par créer les vecteurs composant l'échantillon. Deux décimales suffisent :

```
> n <- 257
> m <- 312
> taille.f <- round(rnorm(m, 165, 6), 2)
> taille.h <- round(rnorm(n, 175, 7), 2)
> poids.f <- round(rnorm(m, 60, 2), 2)
> poids.h <- round(rnorm(n, 75, 4), 2)
```

Passons à la création du tableau de données :

```
> taille <- c(taille.f, taille.h)
> poids <- c(poids.f, poids.h)
> sexe <- rep(c("femme", "homme"), c(m, n))
> donnees <- data.frame(cbind(taille, poids, sexe))
```

La commande `head` est pratique pour vérifier le format des données, en affichant les premières éléments d'un objet selon son type :

```
> head(donnees)

  taille poids  sexe
1 166.25 60.93 femme
2 160.25 58.25 femme
3 159.13 61.09 femme
4 168.28 60.57 femme
5 163.05 59.14 femme
6 159.27 61.7  femme
```

Plaçons les données dans l'itinéraire de recherche pour les manipuler plus aisément :

```
> attach(donnees)
```

Les commandes `by` ou `tapply` appliquent une fonction selon un facteur :

```
> tapply(taille, sexe, min)

femme homme
144.12 158.38

> tapply(taille, sexe, which.min)

femme homme
161 251

> tapply(poids, sexe, min)

femme homme
54.30 64.55

> tapply(poids, sexe, which.min)

femme homme
14 23
```

On peut donc obtenir un résumé statistiques selon chaque groupe en combinant avec `summary` :

```
> summary(donnees)
```

```

      taille      poids      sexe
162.62 : 3   60.32 : 4   femme:312
165.63 : 3   59.02 : 3   homme:257
165.96 : 3   59.08 : 3
171.88 : 3   59.55 : 3
159.46 : 2   60.19 : 3
160.85 : 2   60.34 : 3
(Other):553 (Other):550

> by(taille, sexe, summary)

sexe: femme
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 144.1  161.1   164.6   164.9  168.6   181.9
-----
sexe: homme
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 158.4  169.9   174.7   174.7  179.9   193.5

> by(poids, sexe, summary)

sexe: femme
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  54.30  58.96   60.08   60.06  61.36   66.49
-----
sexe: homme
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  64.55  72.34   75.35   75.40  78.02   86.63

```

Exercice 1.5 (Population de bactéries). Commençons par la fonction générant la population (y compris avec le facteur aléatoire)

```

> PopBacteries <- function(n0,T,p) {
+
+   Na <- rep(0,T)
+   Nb <- rep(0,T)
+   Na[1] <- n0
+
+   for (t in 1:T) {
+     Na[t+1] <- sum(runif(Nb[t]) <=p)
+     Nb[t+1] <- Na[t] + Nb[t]
+   }
+
+   return(list(Na=Na,Nb=Nb,N=Na+Nb))
+ }

```

Voyons quelques exemples

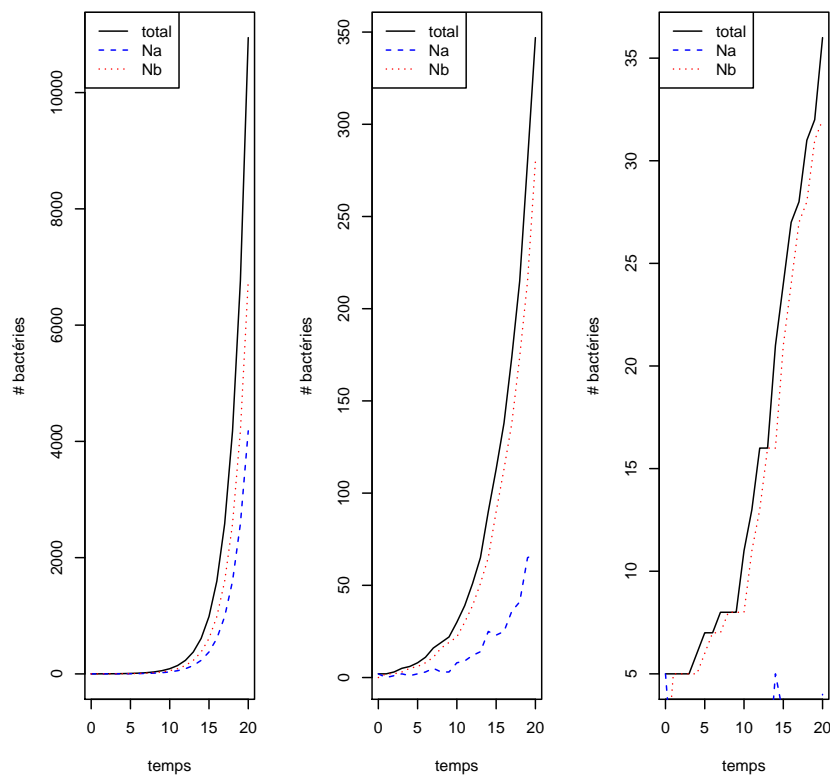
```

> T <- 20
> Pop1 <- PopBacteries(1,T,1)
> Pop2 <- PopBacteries(2,T,0.35)
> Pop3 <- PopBacteries(5,T,0.15)
> par(mfrow=c(1,3))
> plot(0:T,Pop1$N, type="l", col="black",
+      xlab="temps", ylab="# bactéries")
> lines(0:T,Pop1$Na, col="blue", lty=2)

```



```
> lines(0:T,Pop1$Nb, col="red", lty=3)
> legend("topleft",c("total","Na","Nb"),
+       col=c("black","blue","red"), lty=c(1,2,3))
> plot(0:T,Pop2$N, type="l", col="black",
+      xlab="temps", ylab="# bactéries")
> lines(0:T,Pop2$Na, col="blue", lty=2)
> lines(0:T,Pop2$Nb, col="red", lty=3)
> legend("topleft",c("total","Na","Nb"),
+       col=c("black","blue","red"), lty=c(1,2,3))
> plot(0:T,Pop3$N, type="l", col="black",
+      xlab="temps", ylab="# bactéries")
> lines(0:T,Pop3$Na, col="blue", lty=2)
> lines(0:T,Pop3$Nb, col="red", lty=3)
> legend("topleft",c("total","Na","Nb"),
+       col=c("black","blue","red"), lty=c(1,2,3))
```



Par taux d'accroissement d'une population N_t , on entend N_{t+1}/N_t . soit, en moyenne,

```
> mean(Pop1$N[-1]/Pop1$N[-(T + 1)])
[1] 1.602111
> mean(Pop2$N[-1]/Pop2$N[-(T + 1)])
[1] 1.301015
> mean(Pop3$N[-1]/Pop3$N[-(T + 1)])
```

[1] 1.108945

S'il est strictement plus grand que 1, la population augmente ; elle diminue s'il est strictement plus petit ; enfin, elle est stable lorsque le taux d'accroissement vaut exactement 1.

Analyse statistique élémentaire

Exercice 2.1 (Bières).

1. La lecture du fichier se fait à l'aide de la commande `read.csv` :

```
> bieres <- read.csv("bieres.csv")
```

2. Pour peu que le tableau de données soit correctement formaté, c'est-à-dire en mode `factor`, les fonctions `nlevels` et `levels` s'appliquent :

```
> nlevels(bieres$x)
```

```
[1] 5
```

```
> levels(bieres$x)
```

```
[1] "affligem"      "chimay"        "heinecken"    "kronenbourg"  "leffe"
```

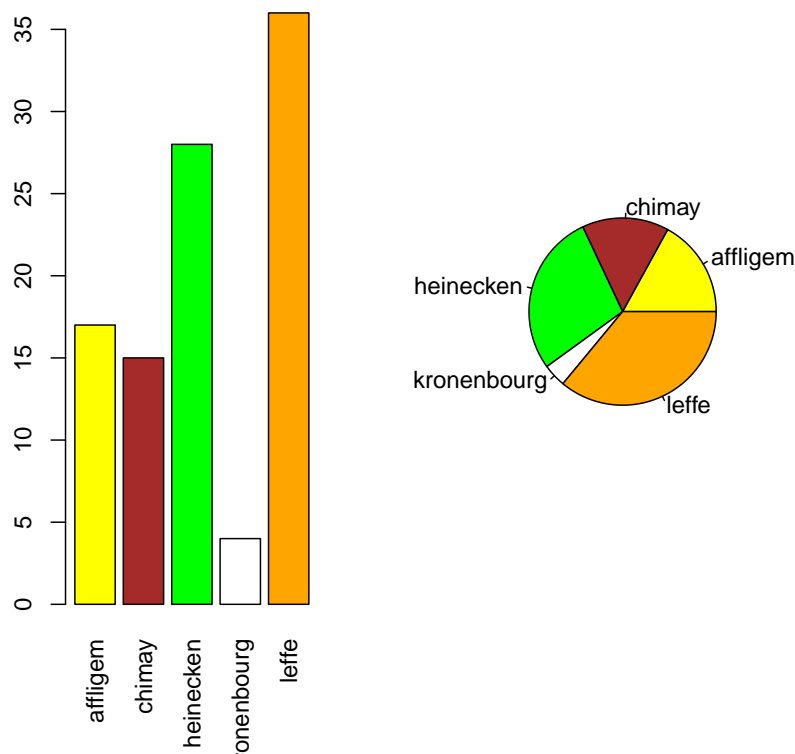
3. De même, fonction `table` est précieuse lors du traitement de données catégorielles.

```
> table(bieres)
```

```
bieres
  affligem    chimay  heinecken kronenbourg    leffe
         17         15         28          4         36
```

Noter l'utilisation des options `las` et `col`, permettant respectivement de déterminer l'orientation du texte le long des axes et la couleurs du tracé.

```
> par(mfrow = c(1, 2))
> barplot(table(bieres), las = 3, col = c("yellow", "brown", "green",
+   "white", "orange"))
> pie(table(bieres), col = c("yellow", "brown", "green", "white",
+   "orange"))
```



Exercice 2.2 (Somnifère).

1. Voyons d'abord la saisie des données (aidez-vous de `scan` pour les tableaux comportant au plus quelques dizaines d'entrées!) :

```
> x <- c(0.7, -1.6, -0.2, -1.2, -0.1, 3.4, 3.7, 0.8, 0, 2, 1.9,
+       0.8, 1.1, 0.1, -0.1, 4.4, 5.5, 1.6, 4.6, 3.4)
```

2. Les opérations entre vecteurs se font par défaut élément par élément :

```
> n <- length(x)
> xbar <- sum(x)/n
> s2 <- sum((x - xbar)^2)/n
> ss2 <- n/(n - 1) * s2
> cat("xbar =", xbar, "s.star^2 =", ss2, "mean(x) =", mean(x),
+     "var(x) =", var(x))
```

```
xbar = 1.54 s.star^2 = 4.072 mean(x) = 1.54 var(x) = 4.072
```

3.

```
> x.s <- sort(x)
> ma.mediane <- mean(c(x.s[floor((n + 1)/2)], x.s[ceiling((n +
+ 1)/2)]))
> cat("Ma médiane = ", ma.mediane, " et median(x) renvoie:",
+     median(x))
```

```
Ma médiane = 0.95 et median(x) renvoie: 0.95
```

4.

```
> stem(x)
```

```

The decimal point is at the |

-0 | 62211
  0 | 01788169
  2 | 0447
  4 | 465

> stem(x, scale = 2)

```

```

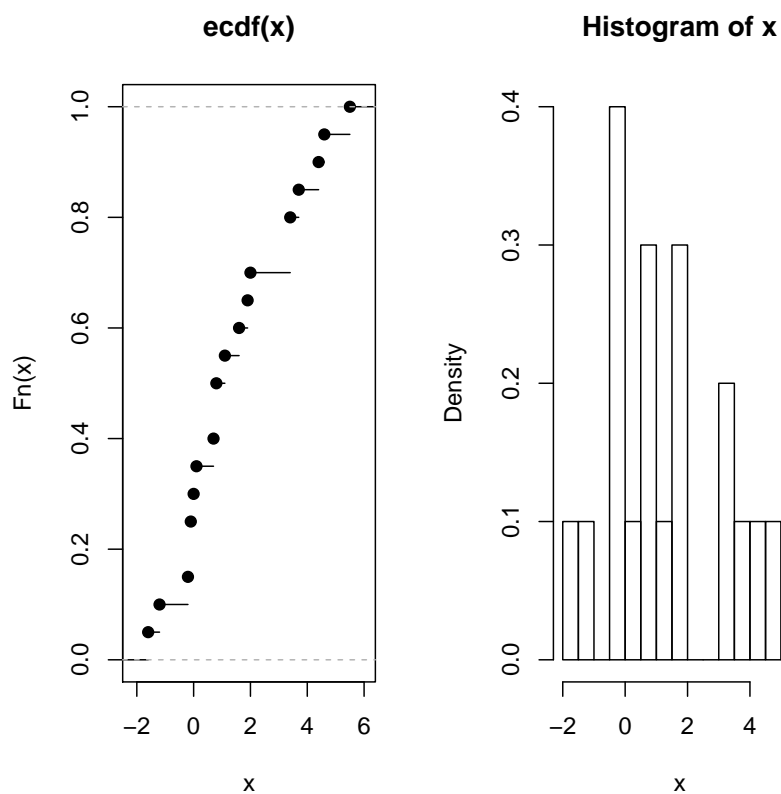
The decimal point is at the |

-1 | 62
-0 | 211
  0 | 01788
  1 | 169
  2 | 0
  3 | 447
  4 | 46
  5 | 5

```

5.

```
> par(mfrow = c(1, 2))
> plot(ecdf(x))
> hist(x, freq = FALSE, nclass = length(unique(x)) + 1)
```



6. Les boîtes à moustaches sont une manière agréable de représenter graphiquement la même information que la commande `summary` appliquée à un tableau de données.

```

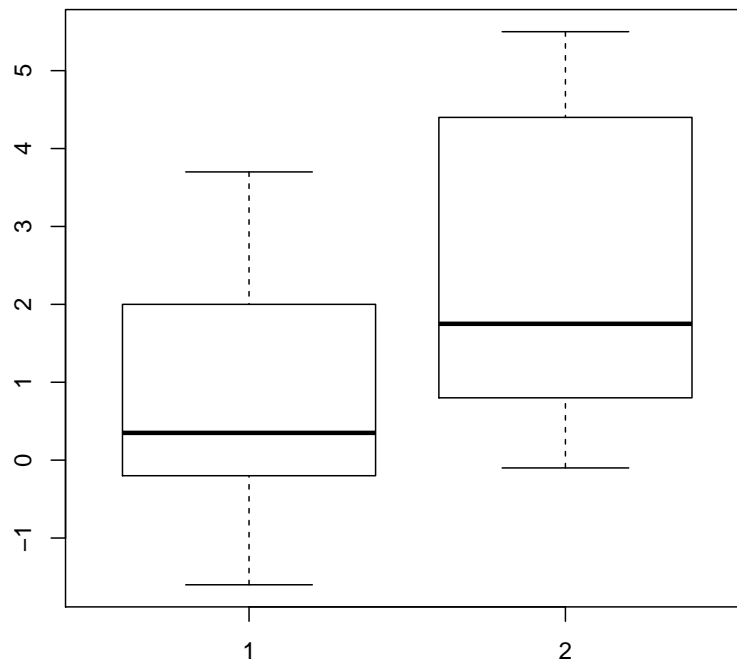
> data <- data.frame(extra = x, group = rep(c(1, 2), each = 10))
> attach(data, warn.conflicts = FALSE)
> by(extra, group, summary)

group: 1
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.600 -0.175   0.350   0.750  1.700   3.700
-----
group: 2
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.100  0.875   1.750   2.330  4.150   5.500

> boxplot(extra ~ group, main = "répartition par groupe")

```

répartition par groupe



Un test de Student ou une analyse de la variance à un facteur permettrait de conclure à l'existence d'un effet groupe.

Ne pas oublier de « détacher » les données de l'itinéraire de recherche pour éviter les mauvaises surprises par la suite :

```
> detach(data)
```

Exercice 2.3 (Iris de Fisher).

1.

```
> data(iris)
> head(iris)
```

```

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1           3.5           1.4           0.2 setosa
2           4.9           3.0           1.4           0.2 setosa
3           4.7           3.2           1.3           0.2 setosa
4           4.6           3.1           1.5           0.2 setosa
5           5.0           3.6           1.4           0.2 setosa
6           5.4           3.9           1.7           0.4 setosa

```

```

> x <- iris[, -5]
> espece <- iris$Species

```

2. La commande `by` permet d'appliquer une fonction à une partition d'un objet :

```

> summary(x)

  Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

> by(x, espece, summary)

espece: setosa
  Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100
1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200
Median :5.000   Median :3.400   Median :1.500   Median :0.200
Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246
3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300
Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600
-----

espece: versicolor
  Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
Min.   :4.900   Min.   :2.000   Min.   :3.000   Min.   :1.000
1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.000   1st Qu.:1.200
Median :5.900   Median :2.800   Median :4.350   Median :1.300
Mean   :5.936   Mean   :2.770   Mean   :4.260   Mean   :1.326
3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.600   3rd Qu.:1.500
Max.   :7.000   Max.   :3.400   Max.   :5.100   Max.   :1.800
-----

espece: virginica
  Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
Min.   :4.900   Min.   :2.200   Min.   :4.500   Min.   :1.400
1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100   1st Qu.:1.800
Median :6.500   Median :3.000   Median :5.550   Median :2.000
Mean   :6.588   Mean   :2.974   Mean   :5.552   Mean   :2.026
3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875   3rd Qu.:2.300
Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500

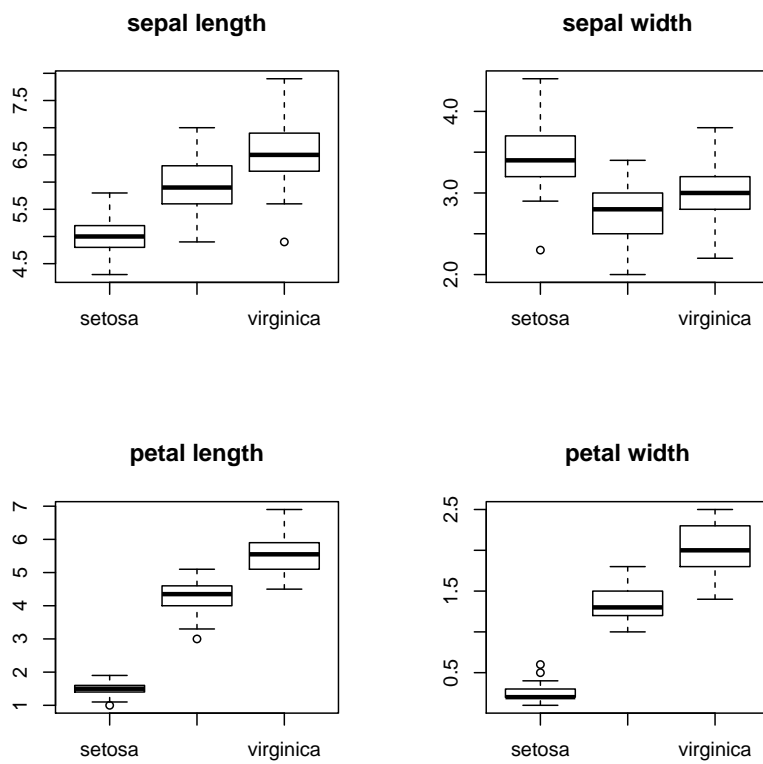
```

3.

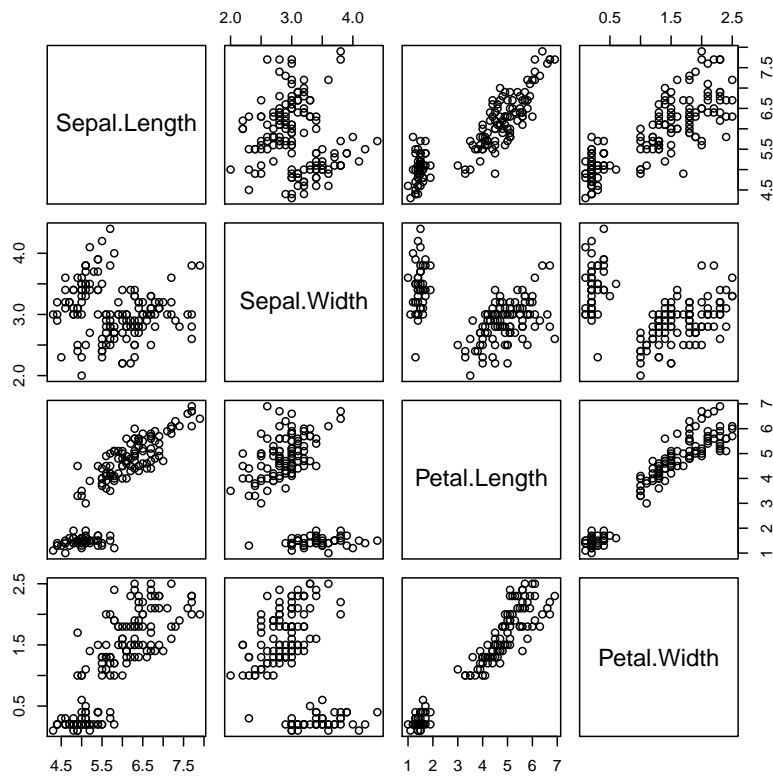
```

> par(mfrow = c(2, 2))
> boxplot(x[, 1] ~ espece, main = "sepal length")
> boxplot(x[, 2] ~ espece, main = "sepal width")
> boxplot(x[, 3] ~ espece, main = "petal length")
> boxplot(x[, 4] ~ espece, main = "petal width")

```



4. `> pairs(x)`



5. `> cov(x)`

```

                Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length  0.68569351 -0.04243400   1.2743154   0.5162707
Sepal.Width   -0.04243400  0.18997942  -0.3296564  -0.1216394
Petal.Length   1.27431544 -0.32965638   3.1162779   1.2956094
Petal.Width    0.51627069 -0.12163937   1.2956094   0.5810063

```

`> cor(x)`

```

                Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   1.0000000  -0.1175698   0.8717538   0.8179411
Sepal.Width   -0.1175698   1.0000000  -0.4284401  -0.3661259
Petal.Length   0.8717538  -0.4284401   1.0000000   0.9628654
Petal.Width    0.8179411  -0.3661259   0.9628654   1.0000000

```

Exercice 2.4 (Puces à ADN).

1. (a) Commençons par créer quelques variables utiles :

```

> n <- 1000
> mu.e <- 1000
> mu.ne <- 400
> sigma.e <- 100
> sigma.ne <- 150

```

(b) Un simple calcul montre que $\mathbb{P}(S^e \leq 700) = \Phi((700 - \mu^e)/\sigma^e)$. On peut vérifier que les tables de la loi normale de R et sous forme papier sont d'accord entre elles :

```
> pnorm((700 - mu.e)/sigma.e)
[1] 0.001349898
```

- (c) On exprime le niveau d'expression d'un gène comme la somme de quatre spots, donc $G^e = 1/4 \sum_{i=1}^4 S_i^e$. La probabilité recherchée est bien évidemment $\mathbb{P}(G^e \leq 700) = \Phi((700 - \mu^e)/(\sigma^e/\sqrt{4}))$, égale à

```
> pnorm((700 - mu.e)/(sigma.e/2))
[1] 9.865876e-10
```

- (d) On cherche la valeur t vérifiant $\mathbb{P}(G^e \leq t) = \mathbb{P}(G^{ne} > t)$, soit,

$$\begin{aligned} \mathbb{P}(G^e \leq t) = 1 - \mathbb{P}(G^{ne} \leq t) &\Leftrightarrow \Phi\left(\frac{t - \mu^e}{\sigma^e/2}\right) = \Phi\left(-\frac{t - \mu^{ne}}{\sigma^{ne}/\sqrt{4}}\right) \\ &\Leftrightarrow \frac{t - \mu^e}{\sigma^e/2} = -\frac{t - \mu^{ne}}{\sigma^{ne}/2}, \end{aligned}$$

d'où $t = 760$

- (e) La probabilité d'un faux négatif est $\mathbb{P}(G^e \leq t)$, soit

```
> t <- 760
> pnorm((t - mu.e)/(sigma.e/2))
[1] 7.933282e-07
```

- (f) La probabilité d'un faux positif est $\mathbb{P}(G^{ne} \geq t)$, soit

```
> t <- 760
> 1 - pnorm((t - mu.ne)/(sigma.ne/2))
[1] 7.933282e-07
```

2. Simulation et graphiques

- (a) Commençons par simuler les populations. Il est utile de repérer l'amplitude des données générées pour pouvoir définir les bornes des axes du graphe par la suite. On stocke donc MIN et le MAX afin de calibrer les axes lors des sorties graphiques.

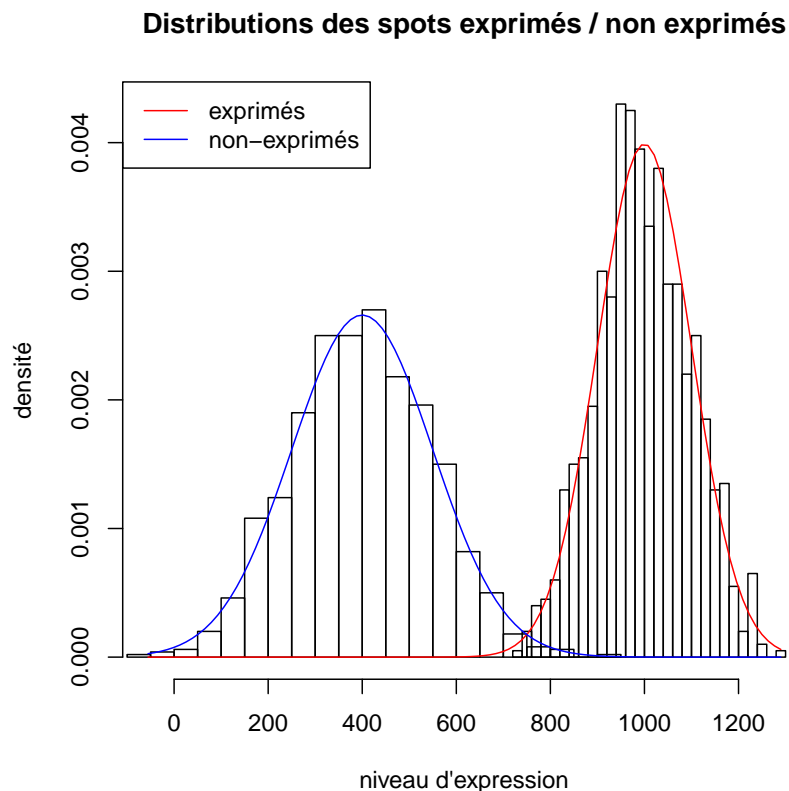
```
> spots.e <- rnorm(n, mu.e, sigma.e)
> spots.ne <- rnorm(n, mu.ne, sigma.ne)
> MIN <- min(spots.e, spots.ne)
> MAX <- max(spots.e, spots.ne)
```

- (b) On stocke les objets histogrammes afin de pouvoir les faire apparaître dans la même fenêtre graphique.

```
> hist.e <- hist(spots.e, nclass = 30, plot = FALSE)
> hist.ne <- hist(spots.ne, nclass = 30, plot = FALSE)
```

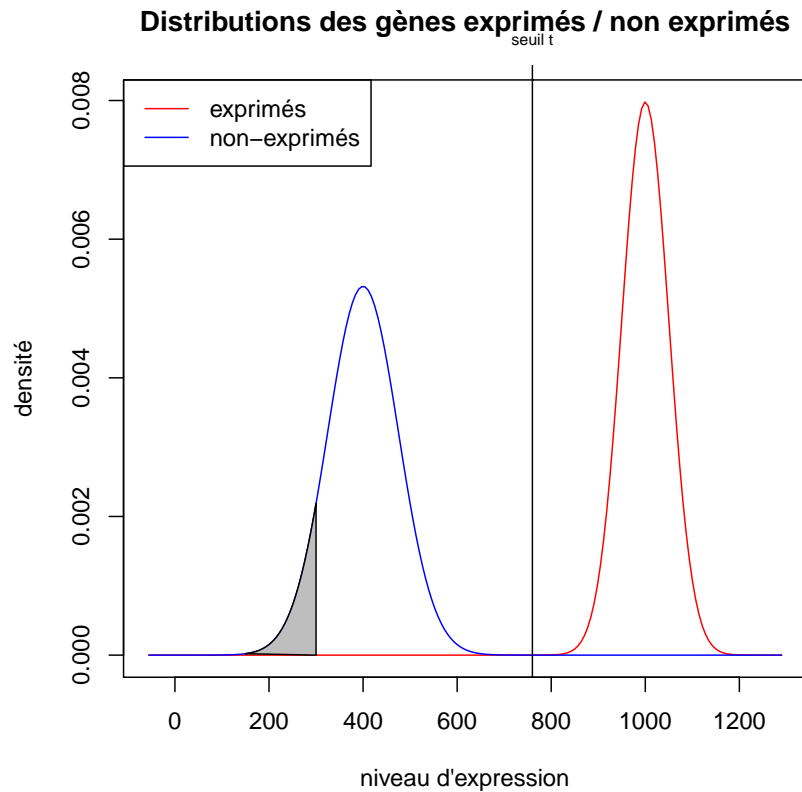
- (c) Noter l'utilisation des fonctions de l'option add et de la fonction curve.

```
> title <- "Distributions des spots exprimés / non exprimés"
> plot(hist.e, freq=FALSE, xlim=c(MIN,MAX), main=title,
+       xlab="niveau d'expression", ylab="densité")
> plot(hist.ne, freq=FALSE, add=TRUE)
> curve(dnorm(x, mu.e, sigma.e), from=MIN, to=MAX,
+       add=TRUE, col="red")
> curve(dnorm(x, mu.ne, sigma.ne), from=MIN, to=MAX,
+       add=TRUE, col="blue")
> legend("topleft", c("exprimés", "non-exprimés"),
+       col=c("red", "blue"), lty=c(1,1))
```



- (d) De prime abord, la fonction polygone peut sembler compliquer à utiliser. En fait, il n'en est rien ! Il suffit de penser à rajouter un point d'ordonnée $y = 0$ car le polygone est tracé en joignant le premier au dernier point... prenez le temps de regarder ce que ça donne lorsque l'on oublie ce point !

```
> title <- "Distributions des gènes exprimés / non exprimés"
> curve(dnorm(x,mu.e,sigma.e/2), n=200,from=MIN,
+       to=MAX,col="red", main=title, ylab="densité",
+       xlab="niveau d'expression")
> curve(dnorm(x,mu.ne,sigma.ne/2), n=200,from=MIN,
+       to=MAX,add=TRUE,col="blue")
> legend("topleft",c("exprimés","non-exprimés"),
+       col=c("red","blue"),lty=c(1,1))
> x <- seq(150,300,len=100)
> y <- dnorm(x,mu.ne,sigma.ne/2)
> x <- c(x,300)
> y <- c(y,0)
> polygon(x,y,col="gray")
> abline(v=t)
> axis(3, at=t, labels="seuil t", cex.axis=0.7)
```



Exercice 2.5 (Maximisation numérique de la vraisemblance).

Questions préliminaires

1. Un peu de calcul permet d'obtenir une écriture sympathique de la log vraisemblance :

$$\begin{aligned} \log L(x_1, \dots, x_n; \mu, \sigma) &= \sum_{i=1}^n \log(f(x_i; \mu, \sigma^2)) \\ &= \sum_{i=1}^n \left(-\log(\sqrt{2\pi\sigma^2}) - \frac{(x_i - \mu)^2}{2\sigma^2} \right) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2. \end{aligned}$$

2. Pour obtenir le maximum d'une fonction concave, il suffit de déterminer quelle valeur annule sa dérivée première. Pour l'espérance, on obtient

$$\begin{aligned} \frac{\partial}{\partial \mu} \log L(x_1, \dots, x_n; \mu, \sigma) = 0 &\Leftrightarrow \sum_{i=1}^n (x_i - \mu) = 0 \\ &\Leftrightarrow \mu = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}. \end{aligned}$$

Pour la variance, on a

$$\begin{aligned}\frac{\partial}{\partial \sigma^2} \log L(x_1, \dots, x_n; \mu, \sigma) &= 0 \\ \Leftrightarrow -\frac{n}{\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^n (x_i - \mu)^2 &= 0 \Leftrightarrow \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2.\end{aligned}$$

En remplaçant μ par son estimateur \bar{x} , on obtient l'estimateur s^2 appelée variance empirique :

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Maximisation numérique

1. Après avoir fixé la valeur des paramètres, on génère une population gaussienne. On calcule la moyenne empirique, la variance empirique et la variance empirique corrigée :

```
> n <- 100
> mu <- pi/2
> sigma <- sqrt(2)
> x <- rnorm(n, mu, sigma)
> X.bar <- sum(x)/n
> S2 <- sum((x - mu)^2)/n
> S2.star <- sum((x - mu)^2)/(n - 1)
```

2. Voici la fonction `loglikelihood` :

```
> loglikelihood <- function(x,mu,sigma) {
+   n <- length(x)
+   L <- -n/2*log(sigma^2*2*pi) -
+     1/(2*sigma^2) * sum((x-mu)^2)
+   return(L)
+ }
```

3. L'optimisation se fait successivement sur chacun des paramètres :

```
> res <- optimize(loglikelihood,x=x,sigma=sigma,
+               maximum=TRUE,lower=-10,upper=10)
> mu.hat <- res$maximum
> res <- optimize(loglikelihood,x=x,mu=mu,
+               maximum=TRUE,lower=0,upper=5)
> sigma.hat <- res$maximum
```

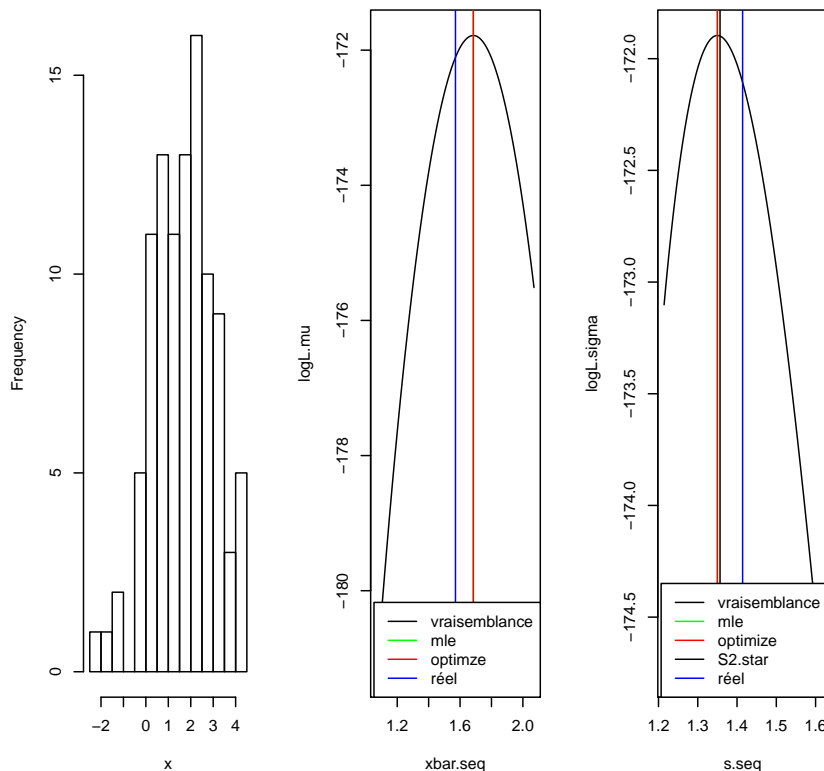
4. Créons les vecteurs pour le tracé des résultats :

```
> logL.mu <- NULL
> logL.sigma <- NULL
> xbar.seq <- seq(from = -0.5 + mu, to = 0.5 + mu, len = 100)
> s.seq <- seq(from = sigma - 0.2, to = sigma + 0.2, len = 100)
> for (xbar in xbar.seq) {
+   logL.mu <- rbind(logL.mu, loglikelihood(x, xbar, sigma))
+ }
> for (s in s.seq) {
+   logL.sigma <- rbind(logL.sigma, loglikelihood(x, mu, s))
+ }
```

Et voici pour les figures :

```
> par(mfrow=c(1,3))
> hist(x,nclass=n/5,main=paste("échantillon normal n=",n))
> plot(xbar.seq,logL.mu,type="l")
> abline(v=c(X.bar,mu.hat,mu),
+        col=c("green","red","blue"))
> legend("bottomright", bty="o", bg="white",
+        c("vraisemblance","mle","optimize","réel"),
+        col=c("black","green","red","blue"),lty=c(1,1,1,1))
> plot(s.seq,logL.sigma,type="l")
> abline(v=c(sqrt(S2),sigma.hat,sqrt(S2.star),sigma),
+        col=c("green","red","black","blue"))
> legend("bottomright", bty="o", bg="white",
+        c("vraisemblance","mle","optimize","S2.star","réel"),
+        col=c("black","green","red","black","blue"),
+        lty=c(1,1,1,1,1))
```

échantillon normal n= 100

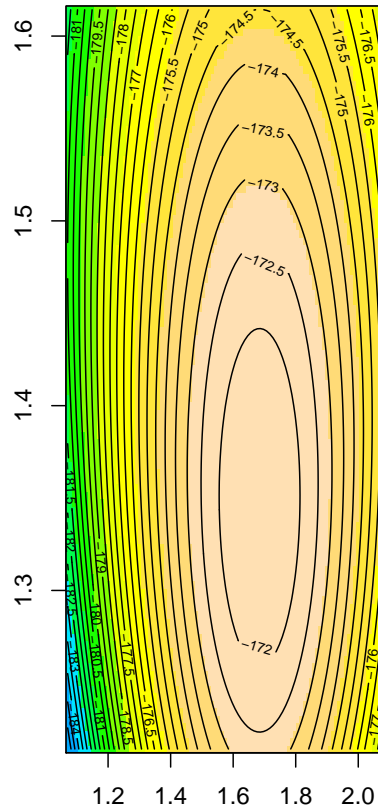
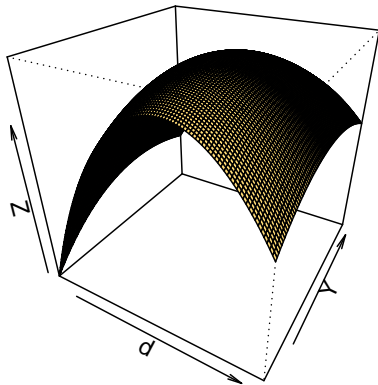


Noter l'utilisation des options `bg` et `box.col` pour éviter que les lignes verticales dues à `abline` ne viennent se superposer aux légendes.

5. À l'aide de la fonction `my2dplot`, cela tient en quelques lignes. Il faut au préalable créer la matrice `logL` pour représenter la surface de la fonction de vraisemblance.

```
> logL <- matrix(0, 100, 100)
> for (i in seq(along = xbar.seq)) {
+   for (j in seq(along = s.seq)) {
+     logL[i, j] <- loglikelihood(x, xbar.seq[i], s.seq[j])
+   }
+ }
```

```
+ }  
+ }  
> d <- list(x = xbar.seq, y = s.seq, z = logL)  
> my2dplot(d)
```



Introduction aux tests d'hypothèses sous R

Exercice 3.1 (H1N1). Voyons d'abord la récupération du tableau de données et son format :

```
> grippe <- read.csv("grippe.csv")
> head(grippe)
```

```
      x
1 [35,65[
2 [18,35[
3 [65,Inf[
4 [18,35[
5 [35,65[
6 [5,18[
```

```
> attach(grippe)
```

On peut maintenant manipuler librement le vecteur x , contenant l'information de classe d'âge pour chaque individu touché. C'est une variables catégorielle : combien y a-t-il de groupes, et combien d'occurrences par groupe ?

```
> nlevels(grippe$x)
```

```
[1] 5
```

```
> levels(grippe$x)
```

```
[1] "[18,35[" "[35,65[" "[5,18[" "[65,Inf[" "]-Inf,5["
```

```
> table(grippe$x)
```

```
 [18,35[ [35,65[ [5,18[ [65,Inf[ ]-Inf,5[
      87      57      47      24      22
```

Calculons les effectifs attendus sous la distribution issue de la première vague (attention au vecteur de probabilités, qui est permuté dans le même ordre que les niveaux, tels que R les classe) :

```
> effectifs.theo <- c(0.35, 0.3, 0.125, 0.15, 0.075) * 237
> names(effectifs.theo) <- levels(grippe$x)
```

La statistique de test du χ^2 est définie par

$$D_{n,k} = \sum_{i=1}^k \frac{(N_{ij} - np_i)^2}{np_i},$$

où N_{ij} sont les effectifs observés et np_i sont les effectifs théoriques ou attendus sous H_0 . La valeur observée de la statistique, notée d_{obs} , se calcule facilement sous R :

```
> d.obs <- sum((table(grippe$x) - effectifs.theo)^2/effectifs.theo)
```

Sous H_0 , $D_{n,k} \sim \chi^2(k-1)$. On rejette l'hypothèse d'égalité des distributions au niveau α si $d_{\text{obs}} > d_{\text{seuil}}$, où $d_{\text{seuil}} = \chi_{k-1;1-\alpha}$. Pour $\alpha = 5\%$ par exemple, on a

```
> alpha <- 0.05
> ddl <- nlevels(grippe$x) - 1
> d.seuil <- qchisq(1 - alpha, ddl)
```

D'où la conclusion :

```
> cat("\nd.obs =", d.obs, " et on rejette pour d.obs >", d.seuil,
+      "au niveau", alpha)
```

d.obs = 17.94113 et on rejette pour d.obs > 9.487729 au niveau 0.05

Voyons la p -valeur :

```
> 1 - pchisq(d.obs, ddl)
```

```
[1] 0.001267223
```

Comparons avec la fonction `chisq.test` :

```
> chisq.test(table(grippe$x), p = c(0.35, 0.3, 0.125, 0.15, 0.075))
```

```
Chi-squared test for given probabilities
```

```
data: table(grippe$x)
X-squared = 17.9411, df = 4, p-value = 0.001267
```

Exercice 3.2 (Alcool au volant). Récupérons les données et voyons le format :

```
> load("accidents.rda")
> head(accidents)
```

```
  blessure alcoolemie
1  légère          >1
2  légère    [0.5,1]
3  légère    [0.5,1]
4  légère          >1
5  grave     [0.5,1]
6  légère    [0.5,1]
```

```
> attach(accidents)
```

Voyons quels sont les niveaux pour chacun des facteurs :

```
> levels(blessure)
```

```
[1] "grave"    "légère"   "mortelle"
```

```
> levels(alcoolemie)
[1] "[0.5,1]" "<0.5" ">1"
```

La commande `table` appliquée à un `data.frame` correctement formaté donne directement les effectifs observés pour le test d'indépendance :

```
> table(accidents)
      alcoolemie
blessure [0.5,1] <0.5 >1
grave    39    49   19
légère   218   526   79
mortelle  18    16   11
```

La table des effectifs attendus sous l'hypothèse d'indépendance se construit comme suit :

```
> n <- nrow(accidents)
> k <- nlevels(blessure)
> l <- nlevels(alcoolemie)
> Ni. <- rowSums(table(accidents))
> N.j <- colSums(table(accidents))
> Ni.N.j <- Ni. %*% t(N.j)
> effectifs.theo <- Ni.N.j/n
> dimnames(effectifs.theo) <- list(levels(blessure), levels(alcoolemie))
> effectifs.theo
      [0.5,1]    <0.5    >1
grave  30.17949  64.85846 11.962051
légère 232.12821 498.86462 92.007179
mortelle 12.69231  27.27692  5.030769
```

On peut maintenant calculer très facilement la valeur observée de la statistique de test, définie par

$$D_{n,k,\ell} = \sum_{i=1}^k \sum_{j=1}^{\ell} \frac{(N_{ij} - N_{i\bullet}N_{\bullet j}/n)^2}{N_{i\bullet}N_{\bullet j}/n},$$

et qui suit une loi du $\chi^2_{(k-1)(\ell-1)}$ sous H_0 , c'est-à-dire sous l'hypothèse d'indépendance des variables. On trouve

```
> d.obs <- sum((table(accidents) - effectifs.theo)^2/effectifs.theo)
```

Le seuil de rejet au niveau α correspond au fractile d'ordre $1 - \alpha$ d'une χ^2 à $(k-1)(\ell-1)$ degrés de liberté. D'où les résultats :

```
> alpha <- 0.05
> ddl <- (k - 1) * (l - 1)
> seuil <- qchisq(1 - alpha, ddl)
> cat("\nd.obs =", d.obs, " et on rejette pour d.obs >", d.seuil,
+     "au niveau", alpha)
d.obs = 28.73554 et on rejette pour d.obs > 9.487729 au niveau 0.05
> 1 - pchisq(d.obs, ddl)
[1] 8.846404e-06
```

ce qui correspond aux résultats de R :

```
> chisq.test(table(accidents))
```

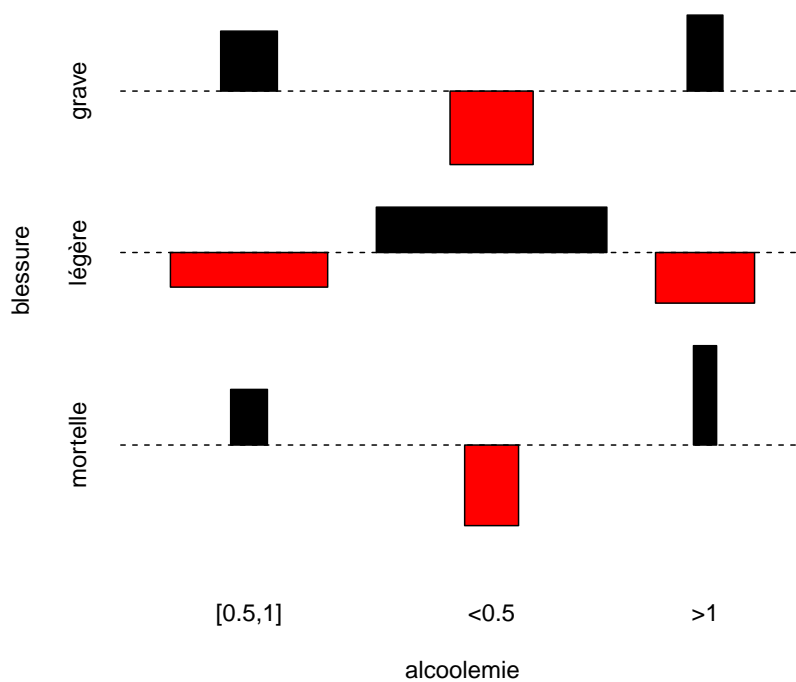
```
Pearson's Chi-squared test
```

```
data: table(accidents)
```

```
X-squared = 28.7355, df = 4, p-value = 8.846e-06
```

Le test est très significatif : taux d'alcoolémie et gravité de la blessure sont des variables dépendantes. Voyons en détail les dépendances via le graphe d'association :

```
> assocplot(t(table(accidents)))
```



Noter que la commande `assocplot` permet une représentation graphique de la déviance de la table de contingence par rapport à l'hypothèse d'indépendance : on y voit clairement que les blessures légères sont corrélées négativement avec les plus fortes consommations d'alcool ; les blessures graves sont au contraire corrélées positivement avec les fortes consommations, etc.

Exercice 3.3 (Pharmacologie).

1. D'abord, la saisie des données

```
> x <- c(15, 14, 21, 12, 17, 12, 19, 18, 20, 21)
> n <- length(x)
> x.bar <- sum(x)/n
> s.2 <- 1/n * sum(x^2) - x.bar^2
> s.star <- sqrt(n/(n - 1) * s.2)
```

2. Commencer par écrire sur le papier les intervalles de confiance unilatères et bilatère pour le test de Student. Il suffit ensuite de discerner les cas selon les valeurs de l'argument `alternative`.

```
> int.conf.mu <- fonction(x, niveau=0.95, alternative = c("two.sided")) {  
+  
+   alpha <- 1-niveau  
+   n <- length(x)  
+   x.bar <- sum(x)/n  
+   s.star <- sqrt( 1/(n-1) * sum(x^2) - sum(x)^2/((n-1)*n) )  
+  
+   if (alternative == "two.sided") {  
+     inf <- x.bar-s.star/sqrt(n)*qt(1-alpha/2,df=n-1)  
+     sup <- x.bar+s.star/sqrt(n)*qt(1-alpha/2,df=n-1)  
+     IC <- paste("[",round(inf,3),";",round(sup,3),"]",sep="")  
+  
+   }  
+   if (alternative == "less") {  
+     inf <- -Inf  
+     sup <- x.bar+s.star/sqrt(n)*qt(1-alpha,df=n-1)  
+     IC <- paste("]",round(inf,3),";",round(sup,3),"]",sep="")  
+   }  
+   if (alternative == "greater") {  
+     inf <- x.bar-s.star/sqrt(n)*qt(1-alpha,df=n-1)  
+     sup <- Inf  
+     IC <- paste("[",round(inf,3),";",round(sup,3),"[]",sep="")  
+   }  
+  
+   return(list(mu=x.bar, IC=IC))  
+ }
```

3. Vérifions que la fonction renvoie bien ce à quoi on s'attend :

```
> int.conf.mu(x, niveau = 0.9)  
  
$mu  
[1] 16.9  
  
$IC  
[1] "[14.884;18.916]"  
  
> int.conf.mu(x, niveau = 0.95)  
  
$mu  
[1] 16.9  
  
$IC  
[1] "[14.412;19.388]"  
  
> int.conf.mu(x, niveau = 0.99)  
  
$mu  
[1] 16.9  
  
$IC  
[1] "[13.325;20.475]"
```

4. Si l'effet est stimulant, aucune raison que l'intervalle soit bilatère :

```
> alpha <- 0.05
> int.conf.mu(x, niveau = 1 - alpha, alternative = "less")
```

```
$mu
[1] 16.9
```

```
$IC
[1] "]-Inf;18.916]"
```

Voyons le seuil à 5% et la p -valeur :

```
> mu0 <- 19
> p.valeur <- pt(sqrt(n) * (x.bar - mu0)/s.star, df = n - 1)
> p.valeur
```

```
[1] 0.0442952
```

```
> seuil <- mu0 - s.star/sqrt(n) * qt(1 - alpha, df = n - 1)
> seuil
```

```
[1] 16.98358
```

```
> t.test(x, mu = mu0, alternative = "less")
```

```
One Sample t-test
```

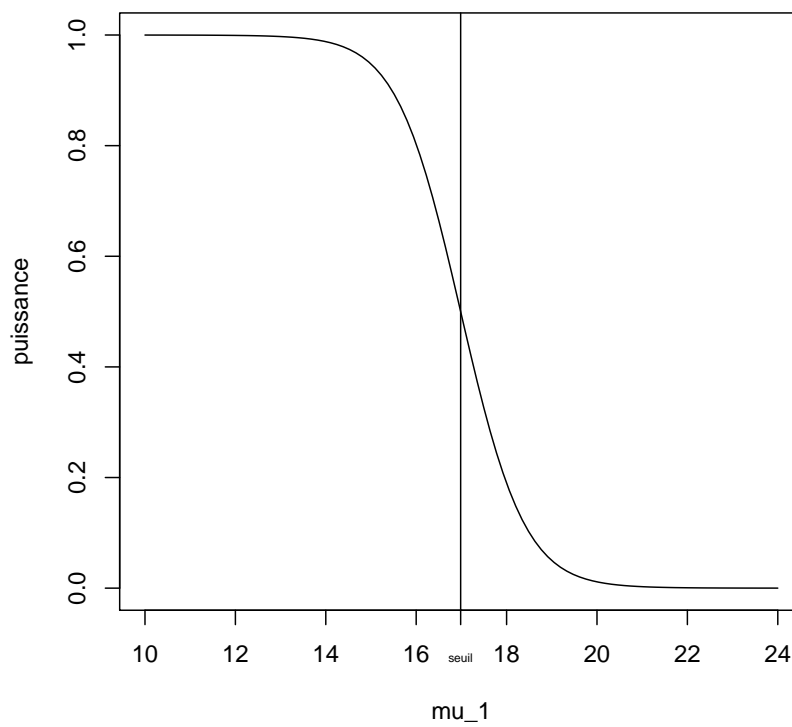
```
data: x
t = -1.9091, df = 9, p-value = 0.04430
alternative hypothesis: true mean is less than 19
95 percent confidence interval:
 -Inf 18.91642
sample estimates:
mean of x
 16.9
```

Le test n'est pas très significatif...

5. La courbe de puissance est facile à produire et permet de trancher sur deux tests prenant des décision similaire (on gardera le plus puissant) :

```
> mu1 <- seq(from = 10, to = 24, length = 100)
> vec <- sqrt(n) * (seuil - mu1)/s.star
> pi <- pt(vec, df = n - 1)
> plot(mu1, pi, xlab = "mu_1", ylab = "puissance", type = "l",
+       main = "courbe de puissance", )
> abline(v = seuil)
> axis(1, at = seuil, labels = "seuil", cex.axis = 0.5)
```

courbe de puissance



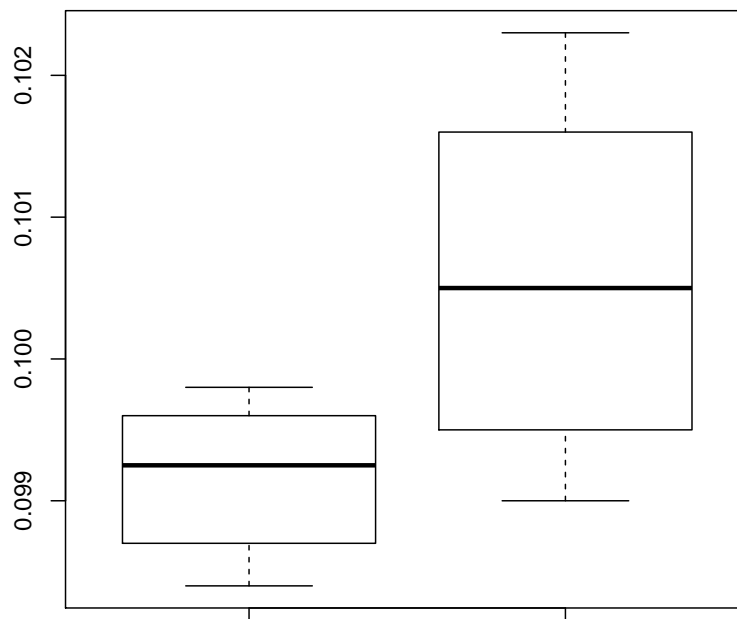
On a également situé le seuil à 5% pour avoir une idée de la puissance du test que nous avons construit : ça n'est pas un test très puissant, puisque l'on se situe autour de $\pi = 50\%$ pour le seuil adopté.

Exercice 3.4 (Pipettes).

1. Après la saisie des données, voyons la répartition par groupe. Il semblerait bien qu'il y ait un effet groupe, mais est-il significatif vu le peu de données ?

```
> pip1 <- c(0.0987, 0.099, 0.0996, 0.0995, 0.0998, 0.0984)
> pip2 <- c(0.1016, 0.1008, 0.1002, 0.0995, 0.099, 0.1023)
> n1 <- length(pip1)
> n2 <- length(pip2)
> boxplot(pip1, pip2, main = "répartition des deux pipettes")
```

répartition des deux pipettes



2. Avant de comparer les espérances, il faut toujours faire un test d'égalité des variances intra groupe puisque c'est l'hypothèse faite par l'analyse de la variance.

```
> F <- var(pip1)/var(pip2)
> alpha <- 0.05
> seuil.1 <- qf(alpha/2, df1 = n1 - 1, df2 = n2 - 1)
> seuil.2 <- qf(1 - alpha/2, df1 = n1 - 1, df2 = n2 - 1)
> p.valeur <- 2 * min(pf(F, df1 = n1 - 1, df2 = n2 - 1), pf(1/F,
+   df1 = n1 - 1, df2 = n2 - 1))
> cat("\nMon test de Fisher\n")
> cat("\nF =", F, "et p.value =", p.valeur)
> cat("\nRejet pour alpha =", alpha, "%: <", seuil.1, ">", seuil.2)
```

Mon test de Fisher

F = 0.1952462 p.value = 0.09733524 Rejet pour F < 0.1399310 > 7.146382

Comparaison avec les sorties de R :


```
> print(var.test(pip1, pip2))

      F test to compare two variances

data:  pip1 and pip2
F = 0.1952, num df = 5, denom df = 5, p-value = 0.09734
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.02732098 1.39530375
sample estimates:
ratio of variances
 0.1952462
```

On conserve donc l'hypothèse d'égalité des variances même si on est proche de la zone de rejet à 5%.

3. Construisons alors le test de Student pour comparer deux espérances (cf. en annexe les rappels de statistiques).

```
> mu.A <- mean(pip1)
> mu.B <- mean(pip2)
> s2 <- ((n1 - 1) * var(pip1) + (n2 - 1) * var(pip2))/(n1 + n2 -
+      2)
> T <- (mu.A - mu.B)/sqrt(s2 * (1/n1 + 1/n2))
> alpha <- 0.05
> seuil <- qt(1 - alpha/2, df = n1 + n2 - 2)
> p.valeur <- 1 - (pt(abs(T), df = n1 + n2 - 2) - pt(-abs(T), df = n1 +
+      n2 - 2))
> cat("Mon test de Student\nT =", T, "et p.value =", p.valeur)
> cat("\nRejet pour alpha =", alpha, "%: abs(T) >", seuil, "\n")
```

```
Mon test de Student
T = -2.502839 et p.value = 0.03129424

Rejet pour alpha = 0.05 %: > 2.228139
```

Le test de R aboutit aux mêmes résultats

```
> t.test(pip1, pip2, var.equal = TRUE)

      Two Sample t-test

data:  pip1 and pip2
t = -2.5028, df = 10, p-value = 0.03129
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.0026463423 -0.0001536577
sample estimates:
mean of x  mean of y
0.09916667 0.10056667
```

4. On peut montrer qu'une analyse de la variance à un facteur entre deux populations est équivalent au test de Student fait ci-dessus.

```
> data <- c(pip1, pip2)
> group <- rep(c(1, 2), each = c(n1, n2))
> print(anova(lm(data ~ group)))
```

Analysis of Variance Table

Response: data

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
group	1	5.8800e-06	5.880e-06	6.2642	0.03129 *
Residuals	10	9.3867e-06	9.387e-07		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Exercice 3.5 (Test de Kolmogorov-Smirnov).

1.

```
> x <- c(160, 330, 600, 780, 1030, 1350, 1950, 2400, 3400, 5000)
> n <- length(x)
> plot(ecdf(x), main = "Fonction de répartition empirique et théorique")
> curve(pexp(x, 1/1700), add = TRUE)
```
2. (a)

```
> Fn <- cumsum(x <= x)/n
> F0 <- pexp(x, 1/1700)
> D <- max(abs(c(0, Fn[-n]) - F0))
> D

[1] 0.09738148
```
- (b) D'après les tables, on ne rejette pas H_0 , ce que confirme l'appelle direct à la fonction de R :

```
> ks.test(x, "pexp", 1/1700)

One-sample Kolmogorov-Smirnov test

data: x
D = 0.0974, p-value = 0.9998
alternative hypothesis: two-sided
```

Analyse de la variance sous R

Exercice 4.1 (Asthme).

1. Visualisation des données

(a) Commençons par charger les données :

```
> asthme <- read.table("asthme.dat", header = TRUE)
> attach(asthme, warn.conflicts = FALSE)
```

(b) Un simple résumé statistique nous indique qu'il existe un effet groupe :

```
> summary(asthme)

      delai      groupe
Min.   :26.00   A:11
1st Qu.:37.00   B:17
Median :45.00   C:15
Mean   :45.60
3rd Qu.:56.00
Max.   :75.00

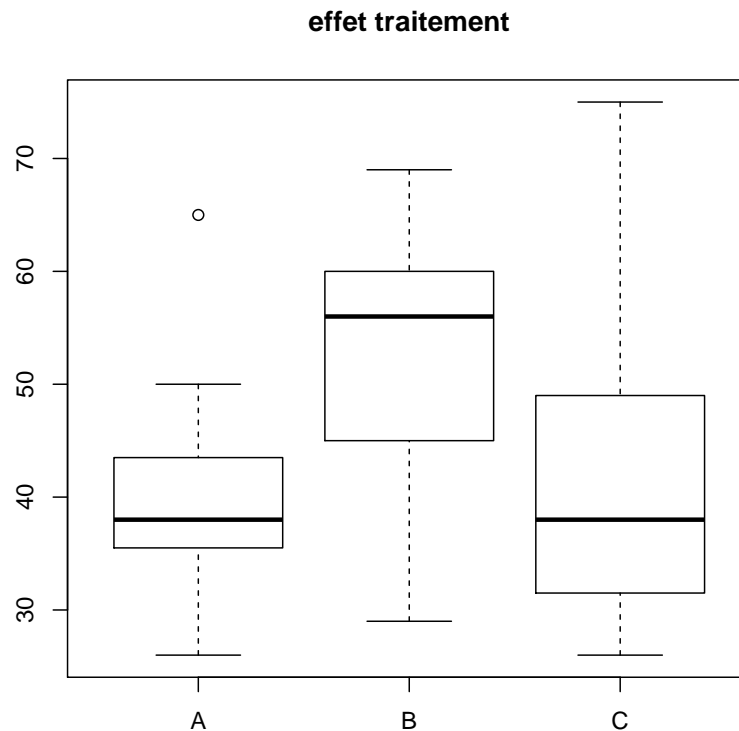
> tapply(delai, groupe, summary)

$A
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
26.00  35.50   38.00   40.27  43.50   65.00

$B
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
29.00  45.00   56.00   52.71  60.00   69.00

$C
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
26.00  31.50   38.00   41.47  49.00   75.00

> boxplot(delai ~ groupe, main = "effet traitement")
```



2. Analyse de la variance

(a) Voici ma fonction `sommes.carres` :

```
> somme.carres <- fonction(donnees,groupe) {
+
+   ## Somme des carrés totale
+   n <- length(donnees)
+   SX <- sum(donnees)
+   SX2 <- sum(donnees^2)
+   SCT <- SX2-SX^2/n
+
+   ## Somme des carrés résiduels
+   nq <- tapply(donnees,groupe,length)
+   SXq <- tapply(donnees,groupe,sum)
+   SX2q <- tapply(donnees^2,groupe,sum)
+   SCRq <- SX2q - SXq^2 / nq
+   SCR <- sum(SCRq)
+
+   ## Somme des carrés des facteurs
+   SCF <- SCT-SCR
+
+   ## degrés de liberté
+   Q <- length(nq)
+   ddl.F <- Q-1
+   ddl.R <- n-Q
+   ddl.T <- n-1
+}
```

```
+ return(list(SCF=SCF,SCR=SCR,SCT=SCT,
+           ddl.F=ddl.F,ddl.R=ddl.R,ddl.T=ddl.T))
+ }
```

- (b) `> SC <- somme.carres(delai, groupe)`
`> CMR <- SCSCR/SCddl.R`
`> CMF <- SCSCF/SCddl.F`
`> R <- CMF/CMR`
`> p.value <- 1 - pf(R, df1 = SC$ddl.F, df2 = SC$ddl.R)`
`> alpha <- 0.05`
`> seuil <- qf(1 - alpha, df1 = SC$ddl.F, df2 = SC$ddl.R)`
`> cat("\nMon analyse de la variance\n R =", R)`
`> cat("\np.value =", p.value, "\nseuil à", alpha, "%:", seuil)`

```
Mon analyse de la variance
R = 5.467381
```

```
p.value = 0.007960023
seuil à 0.05 %: 3.231727
```

- (c) Vérifions que R sort les mêmes résultats :

```
> print(anova(lm(delai ~ groupe)))
```

```
Analysis of Variance Table
```

```
Response: delai
```

```
      Df Sum Sq Mean Sq F value Pr(>F)
groupe  2 1426.8   713.42   5.4674 0.00796 **
Residuals 40 5219.4   130.49
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. Étude de contrastes

Nous aurons besoin des variables suivantes dans ce qui suit :

```
> X. <- tapply(delai, groupe, mean)
> nq <- tapply(delai, groupe, length)
> Q <- length(nq)
> n <- sum(nq)
```

- (a) Testons d'abord $\mu_A = \mu_C$

```
> T <- (X.[[3]] - X.[[1]])/sqrt(CMR * (1/nq[1] + 1/nq[3]))
> alpha <- 0.05
> seuil <- qt(1 - alpha/2, df = n - Q)
> p.value <- 1 - pt(abs(T), df = n - Q) + pt(-abs(T), df = n -
+      Q)
> cat("\nTest muA != muC\nT =", T, "\np.value =", p.value)
> cat("\nseuil à", alpha, "%:", seuil, "\n\n")
```

```
Test muA != muC
```

```
T = 0.2633028
```

```
p.value = 0.7936689
```

```
seuil à 0.05 %: 2.021075
```

Visiblement, μ_A et μ_C ne sont pas significativement différents. Voyons maintenant ce qu'il en est de μ_A et μ_B .

```

> T <- (X.[[2]] - X.[[1]])/sqrt(CMR * (1/nq[1] + 1/nq[2]))
> alpha <- 0.05
> seuil <- qt(1 - alpha/2, df = n - Q)
> p.value <- 1 - pt(abs(T), df = n - Q) + pt(-abs(T), df = n -
+      Q)
> cat("\nTest muA != muB \nT =", T, "\np.value =", p.value)
> cat("\nseuil à", alpha, "%:", seuil)

Test muA != muB
T = 2.812814
p.value = 0.007575904
seuil à 0.05 %: 2.021075

```

Cette fois, la différence entre A et B est significative : le nouveau traitement semble fonctionner.

- (b) Voyons les intervalles de confiance sur le contrastes C_{AB} .

D'abord bilatère

```

> alpha <- 0.05
> IC.inf <- X.[[2]] - X.[[1]] -
+ qt(1-alpha/2,df=n-Q) * sqrt(CMR*(1/nq[1]+1/nq[2]))
> IC.sup <- X.[[2]] - X.[[1]] +
+ qt(1-alpha/2,df=n-Q) * sqrt(CMR*(1/nq[1]+1/nq[2]))
> cat("\nIntervalle de confiance bilatère à =",1-alpha,"%")
> cat("[",IC.inf,";",IC.sup,"]\n")

```

Intervalle de confiance unilatère à = 0.95 %

[3.49963 ; 21.36668 [

puis unilatère :

```

> alpha <- 0.05
> IC.inf <- X.[[2]] - X.[[1]] -
+ qt(1-alpha,df=n-Q) * sqrt(CMR*(1/nq[1]+1/nq[2]))
> IC.sup <- Inf
> cat("\nIntervalle de confiance unilatère à =",1-alpha,"%")
> cat("[",IC.inf,";",IC.sup,"]\n")

```

Intervalle de confiance unilatère à = 0.95 %

[4.990224 ; Inf [

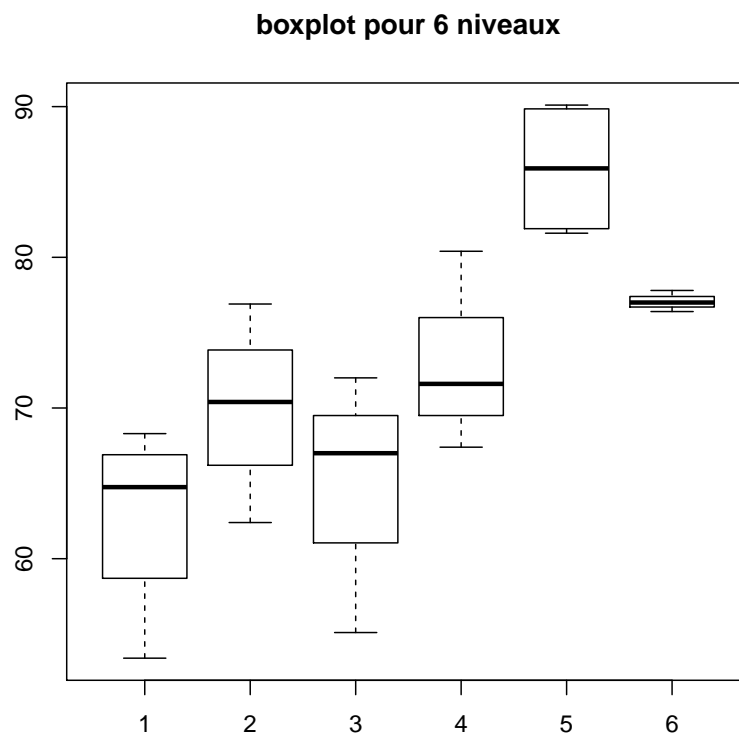
Exercice 4.2 (Rendement de blé).

1. (a) Nous créons un vecteur de données accompagné du vecteur décrivant les groupes associés à chaque rendement :

```
> donnees <- c(53.4, 64, 68.3, 65.5, 76.9, 62.4, 70.8, 70, 55.1,
+ 72, 67, 71.6, 80.4, 67.4, 90.1, 89.6, 81.6, 82.2, 76.4, 77,
+ 77.8)
> groupes <- as.factor(rep(c("1", "2", "3", "4", "5", "6"), c(4,
+ 4, 3, 3, 4, 3)))
```

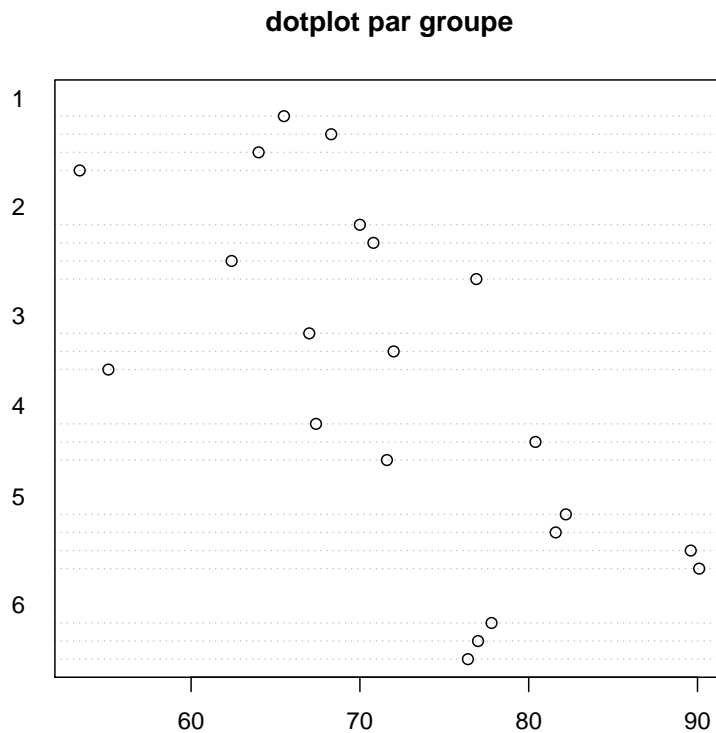
- (b) Voici les boîtes à moustaches : visiblement, les facteurs ont de l'importance, mais ne devrait-on pas les regrouper ?

```
> boxplot(donnees ~ groupes, main = "boxplot pour 6 niveaux")
```



De même pour les nuages de points par groupe :

```
> dotchart(donnees, group = groupes, main = "dotplot par groupe")
```



(c) Allons-y pour le calcul « à la main » des statistiques :

```
> SCT <- sum((donnees - mean(donnees))^2)
> SCR <- sum((donnees - rep(tapply(donnees, groupes, mean), c(4,
+ 4, 3, 3, 4, 3)))^2)
> SCE <- SCT - SCR
> df1 <- nlevels(groupe) - 1
> df2 <- length(donnees) - nlevels(groupe)
> R <- (SCE/df1)/(SCR/df2)
> seuil.1 <- qf(0.01, df1, df2, lower.tail = FALSE)
> seuil.5 <- qf(0.05, df1, df2, lower.tail = FALSE)
> p.value <- pf(R, df1, df2, lower.tail = FALSE)
> cat("\nmon Anova\n R:", R, "p-valeur:", p.value)
```

mon Anova

R: 7.616075 p-valeur: 0.00096833

L'effet des facteurs est très significatif! Comparons avec R :

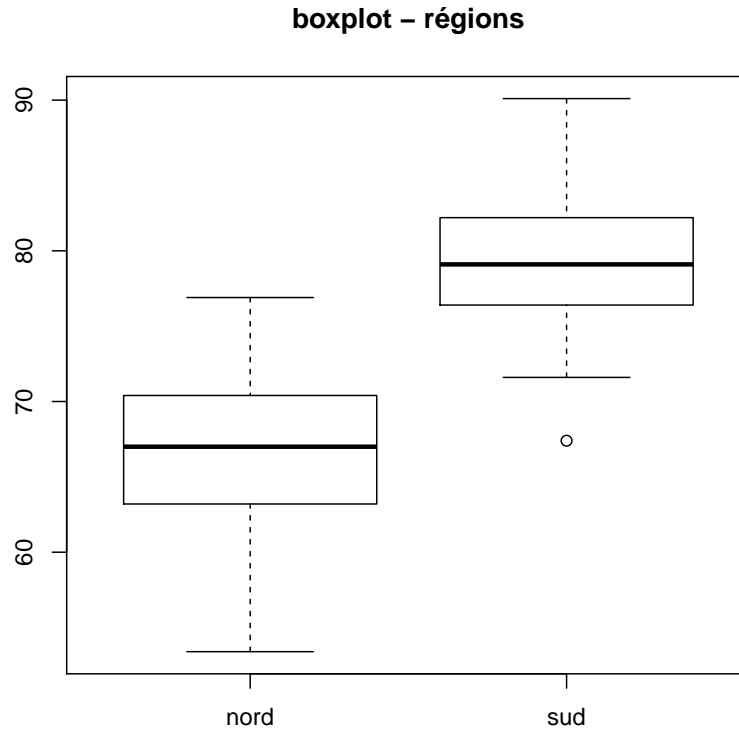
```
> print(summary(aov(donnees ~ groupes)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
groupes	5	1362.28	272.457	7.6161	0.0009683 ***
Residuals	15	536.61	35.774		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- On choisit de diminuer le nombre de niveaux en regroupant les régions, afin de voir si ce modèle, plus simple puisqu'ayant moins de paramètres, reste explicatif :


```
> regions <- as.factor(rep(c("nord", "sud"), c(11, 10)))
> boxplot(donnees ~ regions, main = "boxplot - régions")
> dotchart(donnees, group = regions, main = "dotplot - régions ")
```



```
> print(summary(aov(donnees ~ regions)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
regions	1	949.64	949.64	19.008	0.0003372 ***
Residuals	19	949.26	49.96		

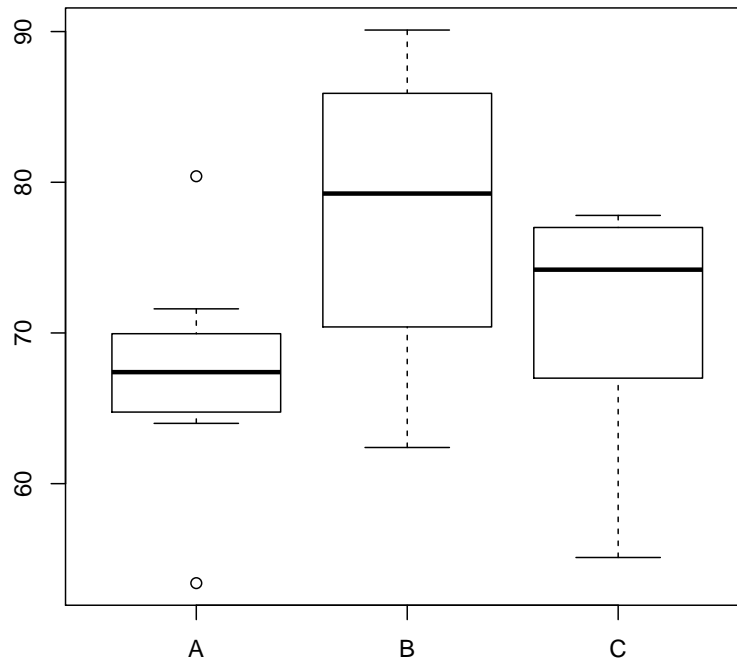
 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

L'effet «régions» est extrêmement significatif!

3. Dans cette question, ce sont les variétés que l'on regroupe :

```
> varietes <- as.factor(rep(c("A", "B", "C", "A", "B", "C"), c(4,
+ 4, 3, 3, 4, 3)))
> boxplot(donnees ~ varietes, main = "boxplot - variétés")
> dotchart(donnees, group = varietes, main = "dot plot - variétés")
```

boxplot – variétés



```
> print(summary(aov(donnees ~ varietes)))
```

```
          Df Sum Sq Mean Sq F value Pr(>F)
varietes   2  447.39  223.694   2.774 0.0891 .
Residuals 18 1451.50   80.639
```

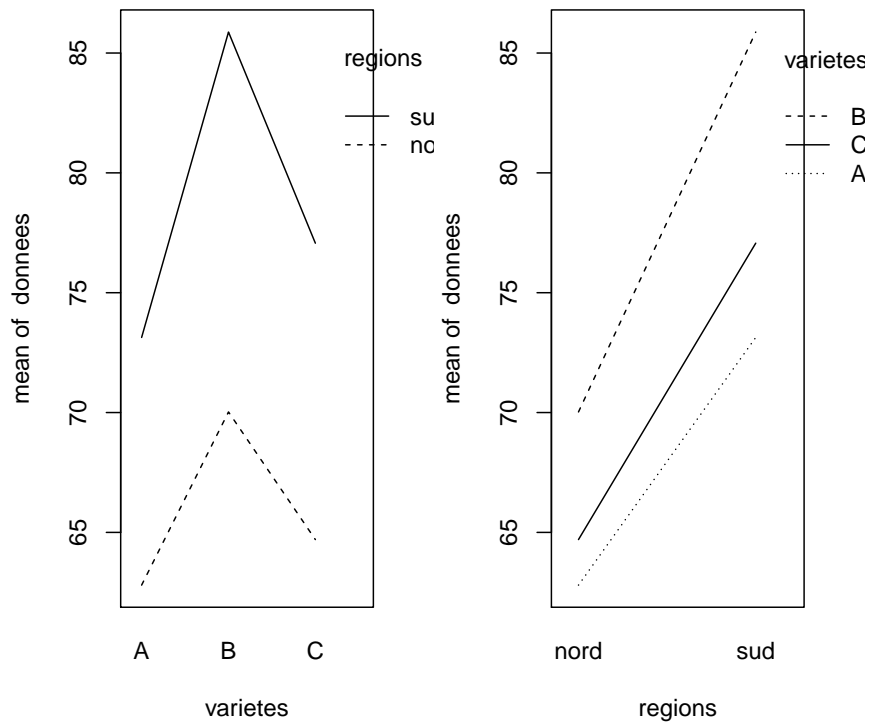
```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On ne peut raisonnablement pas conclure à un effet des variétés... mais dans nos analyses de la variance à un facteur, on néglige systématiquement un facteur pour étudier l'autre. Une anova 2 est plus adaptée.

4. Pour l'anova 2, voyons tout d'abord s'il y a interaction entre les facteurs :

```
> par(mfrow = c(1, 2))
> interaction.plot(varietes, regions, donnees)
> interaction.plot(regions, varietes, donnees)
```



Apparemment non, on peut donc négliger les interactions en proposant un modèle simplement additif :

```
> print(summary(aov(donnees ~ varietes + regions)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
varietes	2	447.39	223.69	6.723	0.00706 **
regions	1	885.86	885.86	26.624	7.853e-05 ***
Residuals	17	565.64	33.27		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Il apparaît que l'effet « variétés » est significatif, mais qu'il était masqué par l'effet « région » lors de l'anova 1.

Introduction au modèle linéaire SOUS R

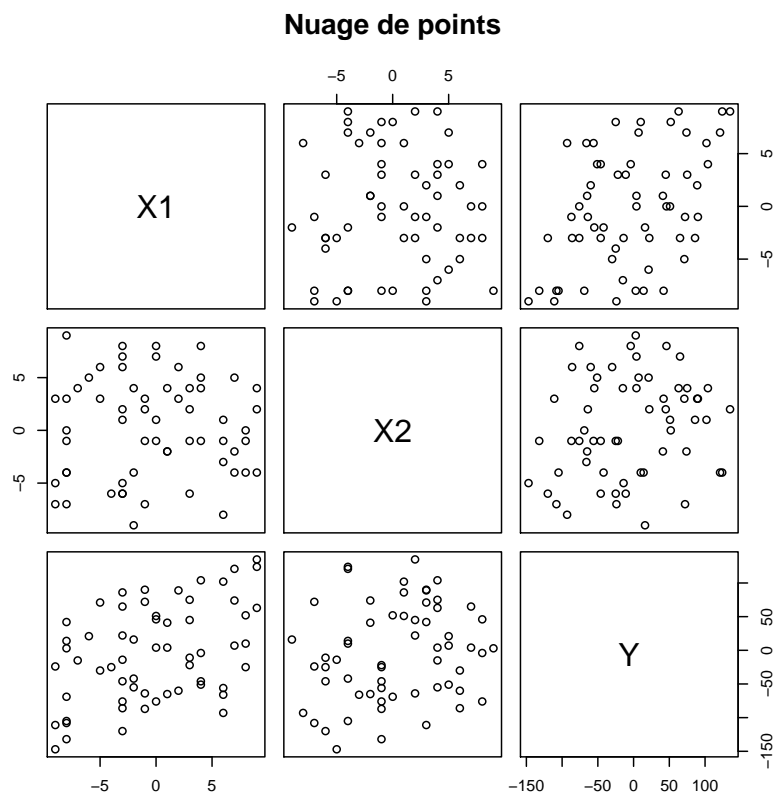
Exercice 5.1 (Performances).

1. Chargeons les données et la bibliothèque `lattice` :

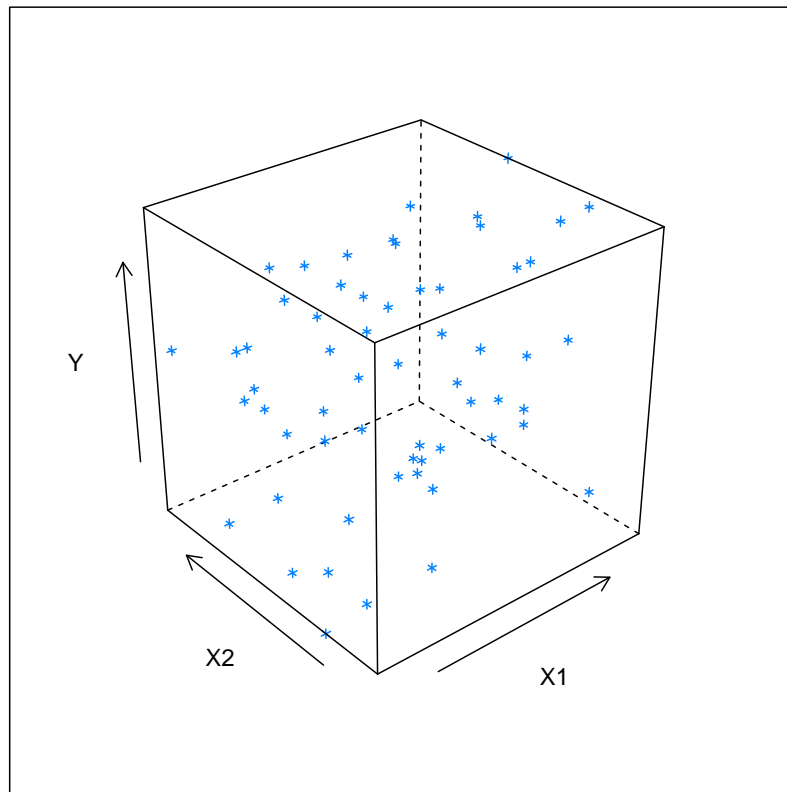
```
> library(lattice)
> load("perf.dat")
> Y <- data$Y
> X1 <- data$X1
> X2 <- data$X2
```

Et voici pour les graphes :

```
> pairs(data,main="Nuage de points")
```



```
> cloud(Y ~ X1 + X2)
```



2. Nous créons quelques variables pour alléger les notations. Il s'agit ensuite de résoudre les équations aux paramètres :

```
> SX1Y <- sum(X1 * Y)
> SX2Y <- sum(X2 * Y)
> SX12 <- sum(X1^2)
> SX22 <- sum(X2^2)
> SX1X2 <- sum(X1 * X2)
> M <- matrix(c(SX12,SX1X2,SX1X2,SX22),2,2)
> b <- c(SX1Y,SX2Y)
> a <- solve(M,b)
> cat("\n les paramètres estimés sont :", a[1], a[2])
```

les paramètres estimés sont : 5.715396 2.883023

Comparons avec R

```
> print(lm(Y ~ X1 + X2 - 1))

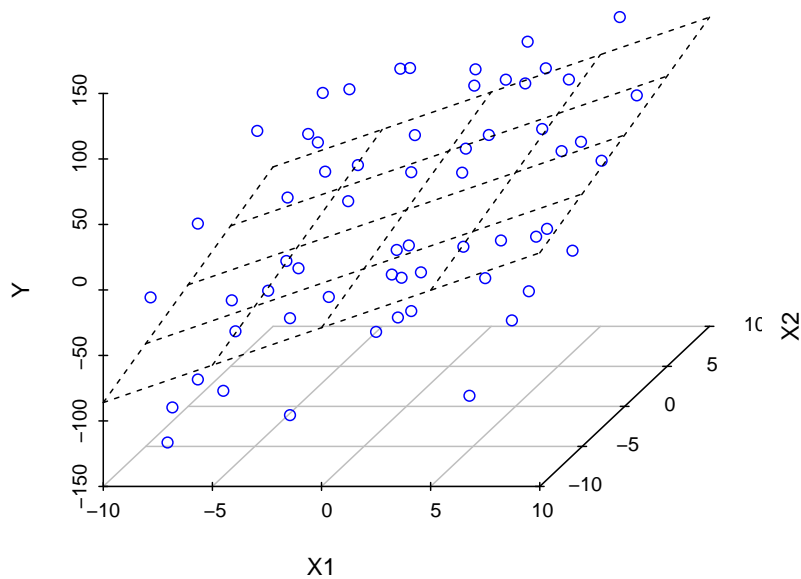
Call:
lm(formula = Y ~ X1 + X2 - 1)

Coefficients:
      X1      X2
5.715  2.883
```

3. La bibliothèque `scatterplot3d` permet de comprendre visuellement la régression en dimension 3 : traçons le plan ajusté sur le nuage de points observé.

```
> library(scatterplot3d)
> s <- scatterplot3d(data, angle=55, color="blue", box=FALSE, main="Résultat de l'estimation")
> s$plane3d(c(0),x.coef=a[1],y.coef=a[2])
```

Résultat de l'estimation



4. Calculons les résidus du modèle H_2 :

```
> Y.hat <- a[1] * X1 + a[2] * X2
> residus <- Y - Y.hat
```

Reste à construire le carré moyen des résidus pour les deux modèles :

```
> SCR.H2 <- sum(residus^2)
> SCR.H0 <- sum(Y^2)
> SCM.2 <- SCR.H0 - SCR.H2
> ddl.H2 <- 58
> ddl.H0 <- 60
> ddl.M2 <- 2
```

On peut alors calculer notre statistique de test et la p -valeur :

```
> R <- (SCM.2/ddl.M2) / (SCR.H2/ddl.H2)
> p.value <- 1-pf(R,2,58)
> cat("\n Valeur de R:", R, "p-value:",p.value)
```

```
Valeur de R: 8.713195 p-value: 0.0004911741
```

Le modèle H_2 est donc approprié par rapport au modèle H_0 . On peut comparer avec R de la sorte :

```

> summary(lm(Y ~ X1 + X2 - 1))

Call:
lm(formula = Y ~ X1 + X2 - 1)

Residuals:
    Min       1Q   Median       3Q      Max
-104.23  -65.21  -11.20   47.74  100.26

Coefficients:
    Estimate Std. Error t value Pr(>|t|)
X1     5.715     1.500   3.809 0.000339 ***
X2     2.883     1.733   1.663 0.101670
---
Signif. codes:  0

```

Cependant la variable X_2 ne semble pas très explicative, d'où l'étude proposée à la question suivante.

5. Voyons le modèle à un paramètre (la droite de régression passant par l'origine) :

```

> mu <- sum(X1 * Y)/sum(X1^2)
> Y.hat <- mu * X1
> residus.H1 <- Y - Y.hat
> SCR.H1 <- sum(residus.H1^2)
> ddl.H1 <- 59
> SCM.1 <- SCR.H0-SCR.H1
> ddl.M1 <- ddl.H0 - ddl.H1
> R.H1 <- (SCM.1/ddl.M1) / (SCR.H1/ddl.H1)
> p.value <- 1-pf(R.H1,1,59)

Valeur de R: 14.23409 p-value: 0.0003764434

```

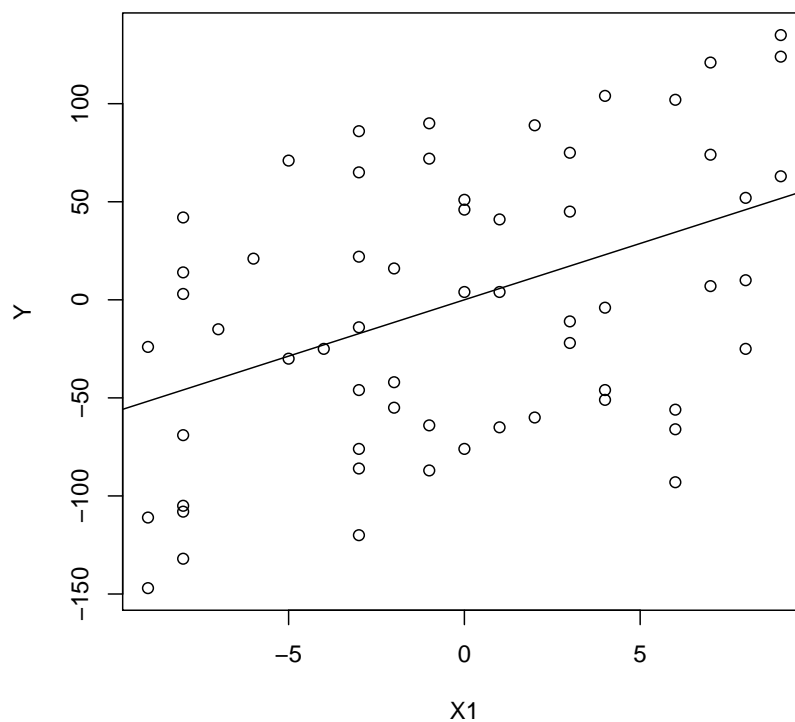
Graphiquement, cela donne :

```

> plot(Y ~ X1, main="modèle à un paramètre")
> abline(a=0,b=mu)

```


modèle à un paramètre



Ce modèle est également pertinent ! Résultats confirmés par R :

```
> summary(lm(Y ~ X1 - 1))
```

```
Call:
lm(formula = Y ~ X1 - 1)
```

```
Residuals:
    Min     1Q   Median     3Q     Max
-127.47  -59.98   -1.51   52.12  103.23
```

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
X1      5.745     1.523   3.773 0.000376 ***
---
Signif. codes:  0
```

Exercice 5.2 (Sida du chat).

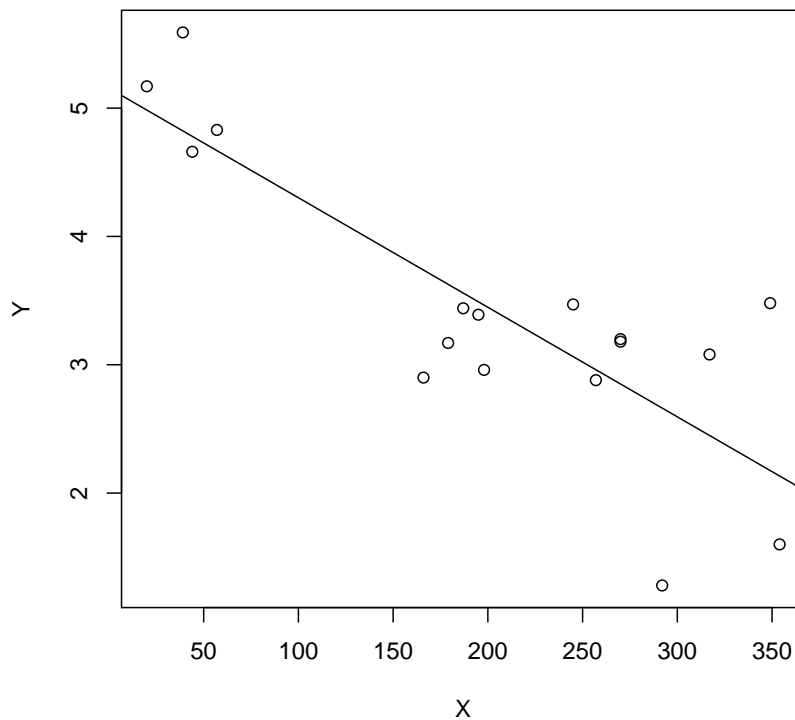
1. Chargeons les données

```
> rm(list=ls())
> load("chat.dat")
> attach(data)
```

On fait l'ajustement de la droite de régression pour les mâles, puis pour les femelles :

```
> attach(males)
> lm.chat.males <- lm(Y ~ X)
> M.males <- matrix(c(sum(X^2),sum(X),sum(X),length(X)),2,2)
> b.males <- c(sum(X*Y),sum(Y))
> a.males <- solve(M.males,b.males)
> plot(Y ~ X, main="droite de regression pour les males")
> abline(a=a.males[2],b=a.males[1])
> Y.hat <- X*a.males[1] + a.males[2]
> r.males <- Y.hat - Y
> SCR.males <- sum(r.males^2)
> CM.males <- SCR.males/(length(X)-2)
> detach(males)
```

droite de regression pour les males

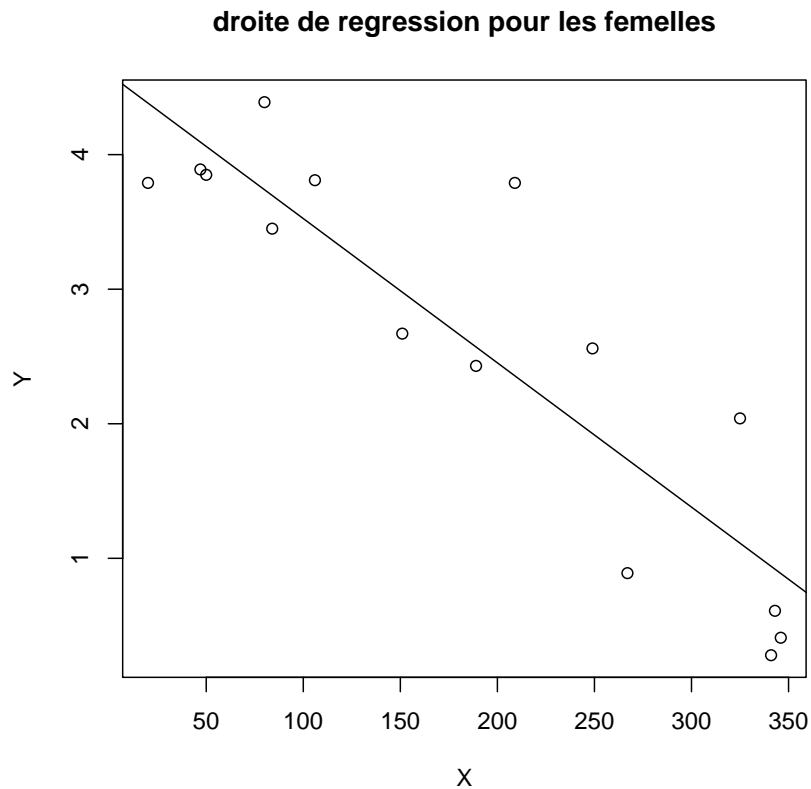


```
> attach(femelles)
> lm.chat.femelles <- lm(Y ~ X)
> M.femelles <- matrix(c(sum(X^2),sum(X),sum(X),length(X)),2,2)
> b.femelles <- c(sum(X*Y),sum(Y))
```

```

> a.femelles <- solve(M.femelles,b.femelles)
> plot(Y ~ X, main="droite de regression pour les femelles")
> abline(a=a.femelles[2],b=a.femelles[1])
> Y.hat <- X*a.femelles[1] + a.femelles[2]
> r.femelles <- Y.hat - Y
> SCR.femelles <- sum(r.femelles^2)
> CM.femelles <- SCR.femelles/(length(X)-2)
> detach(femelles)

```



On peut considérer les variances comme égales :

```

> R.var <- CM.males/CM.femelles
> n1 <- length(males$Y)
> n2 <- length(femelles$Y)
> seuil.1 <- qf(0.95,n1-1,n2-1)
> seuil.2 <- qf(0.05,n1-1,n2-1)
> p.valeur <- 2*min(pf(R.var,df1=n1-1,df2=n2-1),
+                  pf(1/R.var,df1=n1-1,df2=n2-1))

```

R: 0.9547111 seuils: 2.444613 0.421351 p.valeur: 0.9205699

2. Voyons maintenant le modèle mélangé, puisque les variances sont comparables :

```

> X <- c(males$X,femelles$X)
> Y <- c(males$Y,femelles$Y)
> lm.chat <- lm(Y ~ X)
> M <- matrix(c(sum(X^2),sum(X),sum(X),length(X)),2,2)
> b <- c(sum(X*Y),sum(Y))

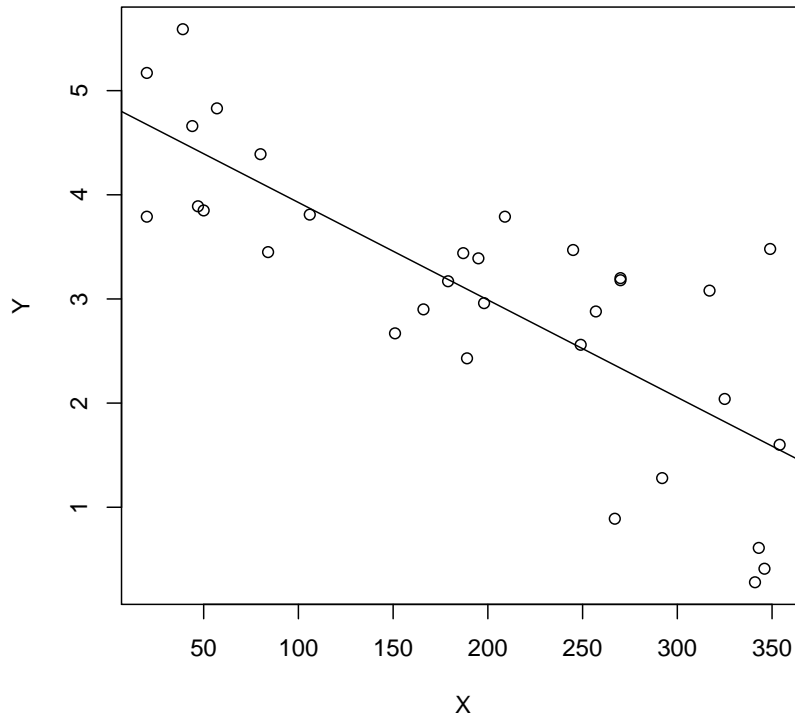
```

```

> a <- solve(M,b)
> plot(Y ~ X, main="droite de regression, melange")
> abline(a=a[2],b=a[1])
> Y.hat <- X*a[1] + a[2]
> r <- Y.hat - Y
> SCR.H2 <- sum(r^2)
> CM.H2 <- SCR.H2/(length(X)-2)

```

droite de regression, melange



On teste H_2 contre H_4 :

```

> SCR.H4 <- SCR.males+SCR.femelles
> CM.H4 <- SCR.H4 / (length(X)-4)
> CMD <- (SCR.H2 - SCR.H4)/2
> R.H2vsH4 <- CMD / CM.H4
> p.value <- 1-pf(R.H2vsH4,2,28)
> cat("R",R.H2vsH4, "p.valeur:",p.value)

```

R 9.11949 p.valeur: 0.0008914503

On rejette donc fortement H_2 pour H_4 : les droites de régression sont différentes de mâle à femelle.

3. Voyons comment se comporte le modèle H_4 face aux modèles où l'on découple le décalage à l'origine de la droite selon mâle/femelle d'une part, et la pente de la droite selon mâle femelle d'autre part. Il s'agit de construire et de résoudre le système linéaire qui régit l'équation aux paramètres. Voici pour le modèle H_a :

```

> Ma <- matrix(c(sum(X^2),sum(males$X),sum(femelles$X),
+               sum(males$X),length(males$X),0,

```

```

+           sum(femelles$X),0,length(femelles$X)),3,3)
> ba <- c(sum(X*Y),sum(males$Y),sum(femelles$Y))
> aa <- solve(Ma,ba)
> Y.hat.males <- males$X*aa[1] + aa[2]
> Y.hat.femelles <- femelles$X*aa[1] + aa[3]
> SCR.Ha <- sum((Y.hat.males - males$Y)^2) + sum((Y.hat.femelles - femelles$Y)^2)
> cat("\nSomme des carrées résiduels pour Ha:", SCR.Ha)

```

Somme des carrées résiduels pour Ha: 12.96104

On aurait pu obtenir directement ces coefficients avec R, travaillant sur la formule du modèle. Au préalable, on définit un `data.frame` adapté. Les résidus sont directement stockés dans l'objet de sortie et sont identiques à ceux calculés « à la main » :

```

> donnees <- data.frame(Y = c(males$Y,femelles$Y), X = c(males$X,femelles$X), sexe = rep(c('M','F'),length(males$X)))
> lm.a <- lm(Y ~ X + sexe - 1, data=donnees)
> cat("SCR de R pour Ha:", sum(lm.a$residuals^2))

```

SCR de R pour Ha: 12.96104

Pour la comparaison de modèles de H_a contre H_4 , on trouve

```

> R.Ha <- (SCR.Ha-SCR.H4)/CM.H4
> pval.Ha <- 1-pf(R.Ha,1,length(X)-4)
> cat("\nR.Ha:",R.Ha,"p.valeur:",pval.Ha)

```

R.Ha: 1.034172 p.valeur: 0.3178879

Cela se passe de manière similaire pour H_b

```

> Mb <- matrix(c(sum(males$X^2),0,sum(males$X),
+               0,sum(femelles$X^2),sum(femelles$X),
+               sum(males$X),sum(femelles$X),length(X)),3,3)
> bb <- c(sum(males$X*males$Y),sum(femelles$X*femelles$Y),sum(Y))
> ab <- solve(Mb,bb)
> Y.hat.males <- males$X*ab[1] + ab[3]
> Y.hat.femelles <- femelles$X*ab[2] + ab[3]
> SCR.Hb <- sum((Y.hat.males - males$Y)^2) + sum((Y.hat.femelles - femelles$Y)^2)
> CM.Hb <- SCR.Hb/(length(X)-3)
> R.Hb <- (SCR.Hb-SCR.H4)/CM.H4
> pval.Hb <- 1-pf(R.Hb,1,length(X)-4)
> cat("\nSCR.Hb:",SCR.Hb,"R.Hb:",R.Hb,"p.valeur:",pval.Hb)

```

SCR.Hb: 13.09977 R.Hb: 1.344943 p.valeur: 0.2559582

La formule à utiliser avec la fonction `lm` de R est

```

> lm.b <- lm(Y ~ X : sexe + 1,data=donnees)
> cat("SCR de R pour Hb:", sum(lm.b$residuals^2))

```

SCR de R pour Hb: 13.09977

Bien que l'on ait rejeté l'hypothèse d'une droite de régression commune (H_4 au profit de H_2), le fait de partager l'un ou l'autre des paramètres de la droite de régression (H_a ou H_b) aboutit à un meilleur modèle que deux droites complètement différentes.

Exercice 5.3. Chargeons le jeu de données `gavotte`

```
> library(faraway)
> rm(list=ls())
> data(gavote)
> head(gavote)
```

```
      equip  econ perAA rural atlanta gore bush other votes ballots
APPLING LEVER  poor 0.182 rural notAtlanta 2093 3940 66 6099 6617
ATKINSON LEVER  poor 0.230 rural notAtlanta 821 1228 22 2071 2149
BACON    LEVER  poor 0.131 rural notAtlanta 956 2010 29 2995 3347
BAKER    OS-CC  poor 0.476 rural notAtlanta 893 615 11 1519 1607
BALDWIN  LEVER  middle 0.359 rural notAtlanta 5893 6041 192 12126 12785
BANKS    LEVER  middle 0.024 rural notAtlanta 1220 3202 111 4533 4773
```

Le « nécessaire » résumé statistique :

```
> summary(gavote)
```

```
      equip      econ      perAA      rural      atlanta
LEVER:74  middle:69  Min.   :0.0000  rural:117  Atlanta  : 15
OS-CC:44  poor   :72  1st Qu.:0.1115  urban: 42  notAtlanta:144
OS-PC:22  rich   :18  Median :0.2330
PAPER: 2          Mean  :0.2430
PUNCH:17         3rd Qu.:0.3480
                   Max.   :0.7650

      gore      bush      other      votes
Min.   : 249  Min.   : 271  Min.   : 5.0  Min.   : 832
1st Qu.: 1386 1st Qu.: 1804 1st Qu.: 30.0 1st Qu.: 3506
Median : 2326 Median : 3597 Median : 86.0 Median : 6299
Mean   : 7020 Mean   : 8929 Mean   : 381.7 Mean   : 16331
3rd Qu.: 4430 3rd Qu.: 7468 3rd Qu.: 210.0 3rd Qu.: 11846
Max.   :154509 Max.   :140494 Max.   :7920.0 Max.   :263211

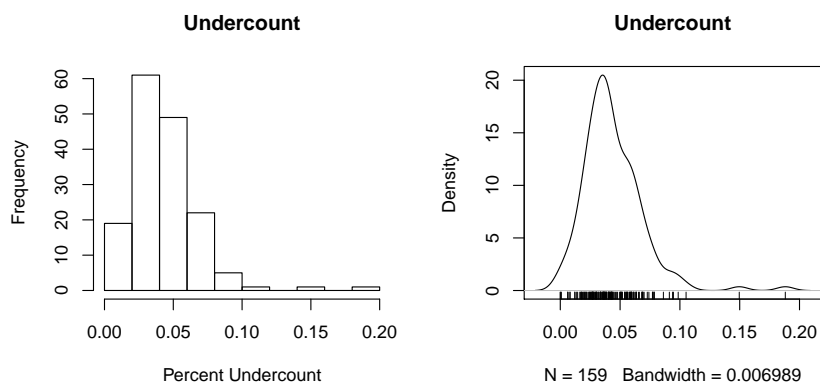
      ballots
Min.   : 881
1st Qu.: 3694
Median : 6712
Mean   : 16927
3rd Qu.: 12251
Max.   :280975
```

Créons `undercount`, qui une variable comptant la proportion de bulletins de vote considérés comme nuls

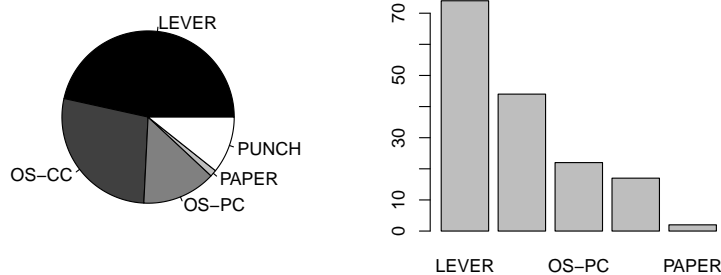
```
> gavote$undercount<- (gavote$ballots - gavote$votes)/gavote$ballots
> summary(gavote$undercount)
```

```
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.02779 0.03983 0.04379 0.05647 0.18810
```

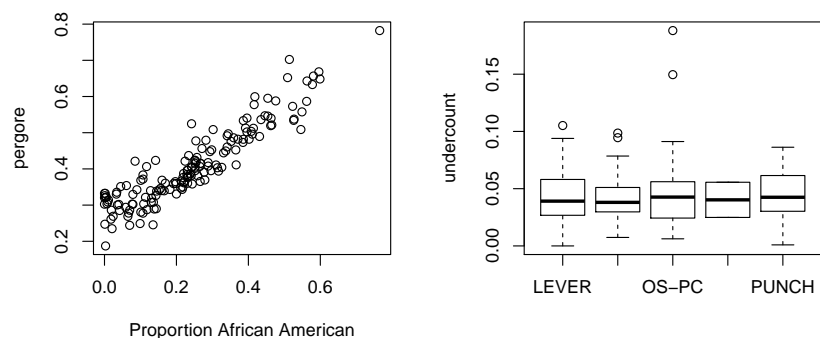
```
> par(mfrow=c(1,2))
> hist(gavote$undercount,main="Undercount",xlab="Percent Undercount")
> plot(density(gavote$undercount),main="Undercount")
> rug(gavote$undercount)
```



```
> par(mfrow=c(1,2))
> pie(table(gavote$equip),col=gray(0:4/4))
> barplot(sort(table(gavote$equip),decreasing=TRUE))
```



```
> par(mfrow=c(1,2))
> gavote$pergore<- gavote$gore/gavote$votes
> plot(pergore ~ perAA , gavote, xlab="Proportion African American")
> plot(undercount ~ equip,gavote,xlab="")
```



Ajustons un modèle de régression linéaire à 1 variable

```
> model1<-lm(undercount~ perAA,gavote)
> summary(model1)
```

```

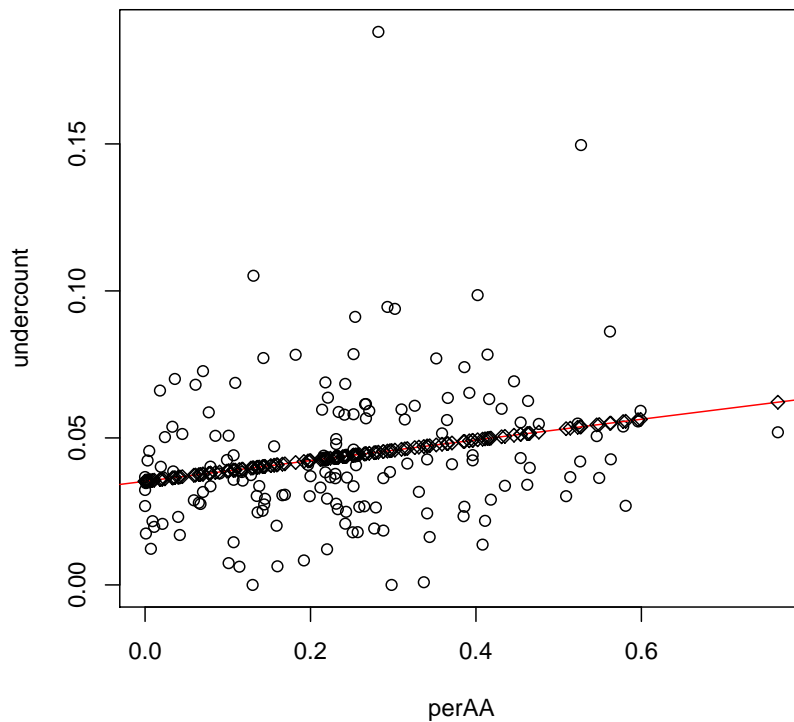
Call:
lm(formula = undercount ~ perAA, data = gavote)

Residuals:
    Min       1Q   Median       3Q      Max
-0.046183 -0.015228 -0.003821  0.012253  0.142957

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.035248   0.003476  10.140 < 2e-16 ***
perAA       0.035162   0.011891   2.957  0.00359 **
---
Signif. codes:  0

> plot(undercount ~ perAA, data= gavote)
> abline(coef(model1), col='red')
> points(predict(model1) ~ perAA, data=gavote, pch=23)

```



Ce code R estime les paramètres du modèle :

$$\text{undercount} = \beta_0 + \beta_2 \text{perAA} + \epsilon$$

Les coefficients du modèle sont données par

```

> coef(model1)

(Intercept)      perAA
0.03524824  0.03516165

```


Les valeurs prédites (pour `undercount`) sont calculés par

```
> model1.prediction<-predict(model1)
> head(model1.prediction)

  APPLING  ATKINSON    BACON    BAKER  BALDWIN  BANKS
0.04164766 0.04333542 0.03985442 0.05198519 0.04787128 0.03609212
```

Les résidus sont données par

```
> model1.residuals<-residuals(model1)
> head(model1.residuals)

  APPLING  ATKINSON    BACON    BAKER  BALDWIN  BANKS
0.036635545 -0.007039473 0.065314388 0.002775232 0.003673501 0.014190718
```

La somme des résidus au carré $RSS = \hat{\epsilon}^t \hat{\epsilon}$

```
> sum(residuals(model1)^2)
```

```
[1] 0.0932819
```

```
> deviance(model1)
```

```
[1] 0.0932819
```

Caclul du R^2 et R^2 ajustés :

```
> n<-length(gavote$undercount)
> print(R2<-1- (deviance(model1)/((n-1)*var(gavote$undercount))))
```

```
[1] 0.05275631
```

```
> p<-2
```

Régressons `undercount` sur `rural` :

```
> summary(model2<-lm(undercount~rural, gavote))
```

Call:

```
lm(formula = undercount ~ rural, data = gavote)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-0.047963 -0.014288 -0.003854  0.010815  0.140158
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.047963    0.002223  21.576 < 2e-16 ***
ruralurban  -0.015791    0.004325  -3.651 0.000355 ***
---
Signif. codes:  0
```

En centrant les variables `pergore` et `perAA`, pour ajuster un modèle (`model3`) de régression linéaire qui explique `undercount` en fonction de `cperAA`, `cpergore`, `rural` et `equip`,

```
> gavote$cpergore <-gavote$pergore - mean(gavote$pergore)
> gavote$cperAA <-gavote$perAA - mean(gavote$perAA)
> model3<-lm(undercount ~ cperAA+cpergore+rural+equip,gavote)
> summary(model3)
```

```

Call:
lm(formula = undercount ~ cperAA + cpergore + rural + equip,
    data = gavote)

Residuals:
    Min       1Q   Median       3Q      Max
-0.059308 -0.013131 -0.002088  0.009017  0.127889

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.043342   0.002823  15.353 < 2e-16 ***
cperAA       0.030004   0.030041   0.999  0.31950
cpergore     0.002785   0.045042   0.062  0.95078
ruralurban  -0.018443   0.004555  -4.049 8.19e-05 ***
equipOS-CC   0.006480   0.004665   1.389  0.16690
equipOS-PC   0.015522   0.005786   2.683  0.00812 **
equipPAPER   -0.008747   0.016805  -0.521  0.60346
equipPUNCH   0.013945   0.006701   2.081  0.03913 *
---
Signif. codes:  0

> step(model3)

Start:  AIC=-1188
undercount ~ cperAA + cpergore + rural + equip

              Df Sum of Sq      RSS      AIC
- cpergore    1 0.0000021 0.081806 -1190.0
- cperAA      1 0.0005404 0.082344 -1189.0
<none>                                0.081804 -1188.0
- equip       4 0.0054478 0.087251 -1185.8
- rural       1 0.0088819 0.090686 -1173.6

Step:  AIC=-1190
undercount ~ cperAA + rural + equip

              Df Sum of Sq      RSS      AIC
<none>                                0.081806 -1190.0
- equip       4 0.0054693 0.087275 -1187.7
- cperAA      1 0.0036702 0.085476 -1185.0
- rural       1 0.0088851 0.090691 -1175.6

Call:
lm(formula = undercount ~ cperAA + rural + equip, data = gavote)

Coefficients:
(Intercept)          cperAA  ruralurban  equipOS-CC  equipOS-PC  equipPAPER
 0.043332      0.031702   -0.018445   0.006503   0.015552   -0.008682
equipPUNCH
 0.013938

```

on obtient un modèle bien meilleur que le modèle `model2`. Comme ce modèle comporte encore deux variables explicatives très corrélés, la procédure `step`, qui cherche à améliorer le modèle au sens AIC, supprime `cpergore`.

En regardant la variable équipement on peut voir que deux équipements en particulier sont associés à l'augmentation du nombre de bulletin nuls (`undercount`). En regroupant ces deux modalités (`PUNCH` et `OS-PC`), on améliore un peu le modèle (il possède moins de paramètres mais la même capacité d'ajustement) :

```
> gavote$OSPC.PUNCH<-as.factor(c(0,0,1,0,1)[as.numeric(gavote$equip)])
> model4<-lm(undercount ~ cperAA+rural+OSPC.PUNCH,gavote)
> summary(model4)
```

Call:

```
lm(formula = undercount ~ cperAA + rural + OSPC.PUNCH, data = gavote)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.059487	-0.012776	-0.001754	0.010592	0.129154

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.045427	0.002274	19.975	< 2e-16	***
cperAA	0.026092	0.011488	2.271	0.02450	*
ruralurban	-0.017816	0.004419	-4.032	8.66e-05	***
OSPC.PUNCH1	0.012522	0.004478	2.796	0.00582	**

Signif. codes: 0

Nous pouvons vérifier que le plus petit modèle est effectivement meilleur (H_0 est le petit modèle et se trouve conservé).

```
> anova(model4,model3)
```

Analysis of Variance Table

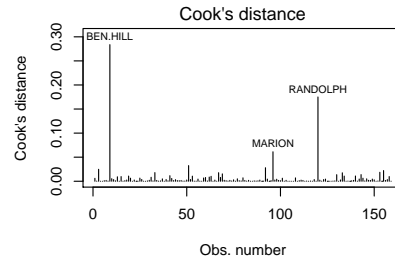
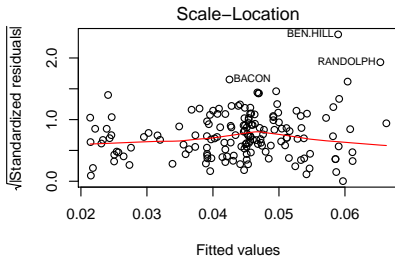
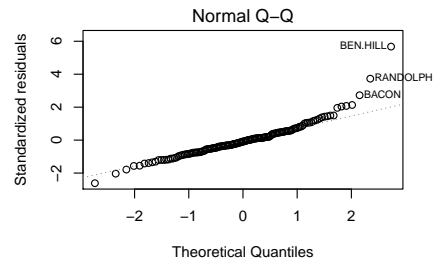
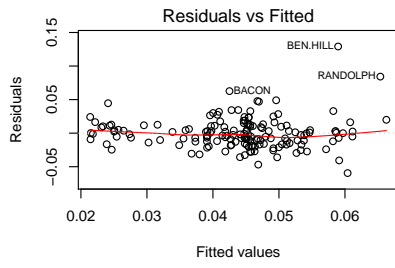
Model 1: undercount ~ cperAA + rural + OSPC.PUNCH

Model 2: undercount ~ cperAA + cpergore + rural + equip

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	155	0.083083				
2	151	0.081804	4	0.0012798	0.5906	0.67

```
> par(mfrow=c(2,2))
```

```
> plot(model4, which = 1:4)
```



Troisième partie

Documents

Bibliographie

- C. Ambroise G.F. McLachlan, K.-A. Do. *Analyzing Microarray Gene Expression Data*. Wiley, 2004.
- E. Paradis. *R pour les débutant*, 2009.
- B. Prum. *Modèle linéaire : comparaison de groupes et régression*. Les Éditions INSERM, 1996.
- R Development Core Team. *R data Import/Export*, v2.10.1 édition, 1999–2009a.
- R Development Core Team. *R Installaiton and Administration*, v2.10.1 édition, 1999–2009b.
- R Development Core Team. *R language definition*, v2.10.1 édition, 1999–2009c.
- R Development Core Team. *Writing R extension*, v2.10.1 édition, 1999–2009d.
- W.N. Venables, D.M. Smith, et the R Development Core Team. *An introduction to R*, v2.10.1 édition, 1999–2009.
- J. Verzani. *simpleR Using R for Introductory Statistics*, 2009.
- B.D. ripley W.N. Venables. *Modern Applied Statistics with S*. Springer, 2002.

Tables statistiques

A.1 Fractiles de la loi du Khi-deux

La table donne les fractiles d'ordre α de la loi du Khi-deux à k degrés de liberté, c'est-à-dire les valeurs de u telles que $\mathbb{P}(\chi^2(k) \geq u) = \alpha$.

$k \setminus \alpha$	0.1	0.05	0.01	0.001
1	2.706	3.841	6.635	10.828
2	4.605	5.991	9.210	13.816
3	6.251	7.815	11.345	16.266
4	7.779	9.488	13.277	18.467
5	9.236	11.070	15.086	20.515
6	10.645	12.592	16.812	22.458
7	12.017	14.067	18.475	24.322
8	13.362	15.507	20.090	26.124
9	14.684	16.919	21.666	27.877
10	15.987	18.307	23.209	29.588
11	17.275	19.675	24.725	31.264
12	18.549	21.026	26.217	32.909
13	19.812	22.362	27.688	34.528
14	21.064	23.685	29.141	36.123
15	22.307	24.996	30.578	37.697
16	23.542	26.296	32.000	39.252
17	24.769	27.587	33.409	40.790
18	25.989	28.869	34.805	42.312
19	27.204	30.144	36.191	43.820
20	28.412	31.410	37.566	45.315
21	29.615	32.671	38.932	46.797
22	30.813	33.924	40.289	48.268
23	32.007	35.172	41.638	49.728
24	33.196	36.415	42.980	51.179
25	34.382	37.652	44.314	52.620
30	40.26	43.77	50.89	59.70
35	46.06	49.80	57.34	66.62
40	51.81	55.76	63.69	73.40
45	57.51	61.66	69.96	80.08
50	63.17	67.50	76.15	86.66
60	74.40	79.08	88.38	99.61
70	85.53	90.53	100.43	112.32
80	96.58	101.88	112.33	124.84
90	107.57	113.15	124.12	137.21
100	118.50	124.34	135.81	149.45

A.2 Fonction de répartition la loi de Gauss centrée-réduite

La table donne les valeurs de $\Phi(x) = \mathbb{P}(\mathcal{N}(0, 1) \leq x)$ où $x = x_1 + x_2$ avec x positif. Pour les valeurs négatives de x , on utilisera la relation $\Phi(x) = 1 - \Phi(-x)$.

$x_1 \setminus x_2$	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.00	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.10	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.20	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.30	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.40	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.50	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.60	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.70	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.80	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.90	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.00	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.10	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.20	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.30	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.40	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.50	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.60	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.70	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.80	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.90	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.00	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.10	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.20	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.30	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.40	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.50	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.60	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.70	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.80	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.90	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.00	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.10	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.20	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.30	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.40	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.50	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
3.60	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.70	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.80	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.90	1	1	1	1	1	1	1	1	1	1

A.3 Fractiles de la loi de Gauss centrée-réduite

La table donne les valeurs de $u_\alpha = \Phi^{-1}(\alpha)$ où $\alpha = \alpha_1 + \alpha_2$. Pour les valeurs de $\alpha < 0.5$, on utilisera la relation $u_\alpha = -u_{1-\alpha}$.

$\alpha_1 \setminus \alpha_2$	0.000	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
0.500	0.0000	0.0025	0.0050	0.0075	0.0100	0.0125	0.0150	0.0175	0.0201	0.0226
0.510	0.0251	0.0276	0.0301	0.0326	0.0351	0.0376	0.0401	0.0426	0.0451	0.0476
0.520	0.0502	0.0527	0.0552	0.0577	0.0602	0.0627	0.0652	0.0677	0.0702	0.0728
0.530	0.0753	0.0778	0.0803	0.0828	0.0853	0.0878	0.0904	0.0929	0.0954	0.0979
0.540	0.1004	0.1030	0.1055	0.1080	0.1105	0.1130	0.1156	0.1181	0.1206	0.1231
0.550	0.1257	0.1282	0.1307	0.1332	0.1358	0.1383	0.1408	0.1434	0.1459	0.1484
0.560	0.1510	0.1535	0.1560	0.1586	0.1611	0.1637	0.1662	0.1687	0.1713	0.1738
0.570	0.1764	0.1789	0.1815	0.1840	0.1866	0.1891	0.1917	0.1942	0.1968	0.1993
0.580	0.2019	0.2045	0.2070	0.2096	0.2121	0.2147	0.2173	0.2198	0.2224	0.2250
0.590	0.2275	0.2301	0.2327	0.2353	0.2378	0.2404	0.2430	0.2456	0.2482	0.2508
0.600	0.2533	0.2559	0.2585	0.2611	0.2637	0.2663	0.2689	0.2715	0.2741	0.2767
0.610	0.2793	0.2819	0.2845	0.2871	0.2898	0.2924	0.2950	0.2976	0.3002	0.3029
0.620	0.3055	0.3081	0.3107	0.3134	0.3160	0.3186	0.3213	0.3239	0.3266	0.3292
0.630	0.3319	0.3345	0.3372	0.3398	0.3425	0.3451	0.3478	0.3505	0.3531	0.3558
0.640	0.3585	0.3611	0.3638	0.3665	0.3692	0.3719	0.3745	0.3772	0.3799	0.3826
0.650	0.3853	0.3880	0.3907	0.3934	0.3961	0.3989	0.4016	0.4043	0.4070	0.4097
0.660	0.4125	0.4152	0.4179	0.4207	0.4234	0.4261	0.4289	0.4316	0.4344	0.4372
0.670	0.4399	0.4427	0.4454	0.4482	0.4510	0.4538	0.4565	0.4593	0.4621	0.4649
0.680	0.4677	0.4705	0.4733	0.4761	0.4789	0.4817	0.4845	0.4874	0.4902	0.4930
0.690	0.4959	0.4987	0.5015	0.5044	0.5072	0.5101	0.5129	0.5158	0.5187	0.5215
0.700	0.5244	0.5273	0.5302	0.5330	0.5359	0.5388	0.5417	0.5446	0.5476	0.5505
0.710	0.5534	0.5563	0.5592	0.5622	0.5651	0.5681	0.5710	0.5740	0.5769	0.5799
0.720	0.5828	0.5858	0.5888	0.5918	0.5948	0.5978	0.6008	0.6038	0.6068	0.6098
0.730	0.6128	0.6158	0.6189	0.6219	0.6250	0.6280	0.6311	0.6341	0.6372	0.6403
0.740	0.6433	0.6464	0.6495	0.6526	0.6557	0.6588	0.6620	0.6651	0.6682	0.6713
0.750	0.6745	0.6776	0.6808	0.6840	0.6871	0.6903	0.6935	0.6967	0.6999	0.7031
0.760	0.7063	0.7095	0.7128	0.7160	0.7192	0.7225	0.7257	0.7290	0.7323	0.7356
0.770	0.7388	0.7421	0.7454	0.7488	0.7521	0.7554	0.7588	0.7621	0.7655	0.7688
0.780	0.7722	0.7756	0.7790	0.7824	0.7858	0.7892	0.7926	0.7961	0.7995	0.8030
0.790	0.8064	0.8099	0.8134	0.8169	0.8204	0.8239	0.8274	0.8310	0.8345	0.8381
0.800	0.8416	0.8452	0.8488	0.8524	0.8560	0.8596	0.8633	0.8669	0.8705	0.8742
0.810	0.8779	0.8816	0.8853	0.8890	0.8927	0.8965	0.9002	0.9040	0.9078	0.9116
0.820	0.9154	0.9192	0.9230	0.9269	0.9307	0.9346	0.9385	0.9424	0.9463	0.9502
0.830	0.9542	0.9581	0.9621	0.9661	0.9701	0.9741	0.9782	0.9822	0.9863	0.9904
0.840	0.9945	0.9986	1.0027	1.0069	1.0110	1.0152	1.0194	1.0237	1.0279	1.0322
0.850	1.0364	1.0407	1.0450	1.0494	1.0537	1.0581	1.0625	1.0669	1.0714	1.0758
0.860	1.0803	1.0848	1.0893	1.0939	1.0985	1.1031	1.1077	1.1123	1.1170	1.1217
0.870	1.1264	1.1311	1.1359	1.1407	1.1455	1.1503	1.1552	1.1601	1.1650	1.1700
0.880	1.1750	1.1800	1.1850	1.1901	1.1952	1.2004	1.2055	1.2107	1.2160	1.2212
0.890	1.2265	1.2319	1.2372	1.2426	1.2481	1.2536	1.2591	1.2646	1.2702	1.2759
0.900	1.2816	1.2873	1.2930	1.2988	1.3047	1.3106	1.3165	1.3225	1.3285	1.3346
0.910	1.3408	1.3469	1.3532	1.3595	1.3658	1.3722	1.3787	1.3852	1.3917	1.3984
0.920	1.4051	1.4118	1.4187	1.4255	1.4325	1.4395	1.4466	1.4538	1.4611	1.4684
0.930	1.4758	1.4833	1.4909	1.4985	1.5063	1.5141	1.5220	1.5301	1.5382	1.5464
0.940	1.5548	1.5632	1.5718	1.5805	1.5893	1.5982	1.6072	1.6164	1.6258	1.6352
0.950	1.6449	1.6546	1.6646	1.6747	1.6849	1.6954	1.7060	1.7169	1.7279	1.7392
0.960	1.7507	1.7624	1.7744	1.7866	1.7991	1.8119	1.8250	1.8384	1.8522	1.8663
0.970	1.8808	1.8957	1.9110	1.9268	1.9431	1.9600	1.9774	1.9954	2.0141	2.0335
0.980	2.0537	2.0749	2.0969	2.1201	2.1444	2.1701	2.1973	2.2262	2.2571	2.2904
0.990	2.3263	2.3656	2.4089	2.4573	2.5121	2.5758	2.6521	2.7478	2.8782	3.0902

A.4 Fractiles de la loi de Student

La table donne les fractiles d'ordre α de la loi de Student à ν degrés de liberté, c'est-à-dire les valeurs de t telles que $\mathbb{P}(\mathcal{T}_\nu \leq t) = \alpha$.

$\nu \backslash \alpha$	0.60	0.70	0.80	0.90	0.95	0.975	0.990	0.995	0.999	0.9995
1	0.325	0.727	1.376	3.078	6.314	12.71	31.82	63.66	318.3	636.6
2	0.289	0.617	1.061	1.886	2.920	4.303	6.965	9.925	22.33	31.60
3	0.277	0.584	0.978	1.638	2.353	3.182	4.541	5.841	10.22	12.94
4	0.271	0.569	0.941	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.267	0.559	0.920	1.476	2.015	2.571	3.365	4.032	5.893	6.859
6	0.265	0.553	0.906	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.263	0.549	0.896	1.415	1.895	2.365	2.998	3.499	4.785	5.405
8	0.262	0.546	0.889	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.261	0.543	0.883	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.260	0.542	0.879	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.260	0.540	0.876	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.259	0.539	0.873	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.259	0.538	0.870	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.258	0.537	0.868	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.258	0.536	0.866	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.258	0.535	0.865	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.257	0.534	0.863	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.257	0.534	0.862	1.330	1.734	2.101	2.552	2.878	3.611	3.922
19	0.257	0.533	0.861	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.257	0.533	0.860	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.257	0.532	0.859	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.256	0.532	0.858	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.256	0.532	0.858	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	0.256	0.531	0.857	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.256	0.531	0.856	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.256	0.531	0.856	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.256	0.531	0.855	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.256	0.530	0.855	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.256	0.530	0.854	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.256	0.530	0.854	1.310	1.697	2.042	2.457	2.750	3.385	3.646
32	0.256	0.530	0.853	1.309	1.694	2.037	2.449	2.738	3.365	3.622
34	0.255	0.529	0.852	1.307	1.691	2.032	2.441	2.728	3.348	3.601
36	0.255	0.529	0.852	1.306	1.688	2.028	2.434	2.719	3.333	3.582
38	0.255	0.529	0.851	1.304	1.686	2.024	2.429	2.712	3.319	3.566
40	0.255	0.529	0.851	1.303	1.684	2.021	2.423	2.704	3.307	3.551
50	0.255	0.528	0.849	1.298	1.676	2.009	2.403	2.678	3.261	3.496
60	0.254	0.527	0.848	1.296	1.671	2.000	2.390	2.660	3.232	3.460
70	0.254	0.527	0.847	1.294	1.667	1.994	2.381	2.648	3.211	3.435
80	0.254	0.527	0.846	1.292	1.664	1.990	2.374	2.639	3.195	3.415
90	0.254	0.526	0.846	1.291	1.662	1.987	2.368	2.632	3.183	3.402
100	0.254	0.526	0.845	1.290	1.660	1.984	2.365	2.626	3.174	3.389
200	0.254	0.525	0.843	1.286	1.653	1.972	2.345	2.601	3.131	3.339
500	0.253	0.525	0.842	1.283	1.648	1.965	2.334	2.586	3.106	3.310
∞	0.253	0.524	0.842	1.282	1.645	1.960	2.326	2.576	3.090	3.291

A.5 Fractiles de la loi de Fisher

$\nu_2 \backslash \nu_1$	1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	∞
1	39.86	49.5	53.6	55.83	57.24	58.2	58.9	59.44	59.86	60.2	60.7	61.2	61.7	62.0	62.26	62.53	62.8	66.12
2	8.53	9.00	9.16	9.24	9.29	9.33	9.35	9.37	9.38	9.39	9.41	9.42	9.44	9.45	9.46	9.47	9.47	9.49
3	5.54	5.46	5.39	5.34	5.31	5.28	5.27	5.25	5.24	5.23	5.22	5.20	5.18	5.18	5.17	5.16	5.15	5.13
4	4.54	4.32	4.19	4.11	4.05	4.01	3.98	3.95	3.94	3.92	3.90	3.87	3.84	3.83	3.82	3.80	3.79	3.76
5	4.06	3.78	3.62	3.52	3.45	3.40	3.37	3.34	3.32	3.30	3.27	3.24	3.21	3.19	3.17	3.16	3.14	3.10
6	3.78	3.46	3.29	3.18	3.11	3.05	3.01	2.98	2.96	2.94	2.90	2.87	2.84	2.82	2.80	2.78	2.76	2.72
7	3.59	3.26	3.07	2.96	2.88	2.83	2.78	2.75	2.72	2.70	2.67	2.63	2.59	2.58	2.56	2.54	2.51	2.47
8	3.46	3.11	2.92	2.81	2.73	2.67	2.62	2.59	2.56	2.54	2.50	2.46	2.42	2.40	2.38	2.36	2.34	2.29
9	3.36	3.01	2.81	2.69	2.61	2.55	2.51	2.47	2.44	2.42	2.38	2.34	2.30	2.28	2.25	2.23	2.21	2.16
10	3.29	2.92	2.73	2.61	2.52	2.46	2.41	2.38	2.35	2.32	2.28	2.24	2.20	2.18	2.16	2.13	2.11	2.06
11	3.23	2.86	2.66	2.54	2.45	2.39	2.34	2.30	2.27	2.25	2.21	2.17	2.12	2.10	2.08	2.05	2.03	1.97
12	3.18	2.81	2.61	2.48	2.39	2.33	2.28	2.24	2.21	2.19	2.15	2.10	2.06	2.04	2.01	1.99	1.96	1.90
13	3.14	2.76	2.56	2.43	2.35	2.28	2.23	2.20	2.16	2.14	2.10	2.05	2.01	1.98	1.96	1.93	1.90	1.85
14	3.10	2.73	2.52	2.39	2.31	2.24	2.19	2.15	2.12	2.10	2.05	2.01	1.96	1.94	1.91	1.89	1.86	1.80
15	3.07	2.70	2.49	2.36	2.27	2.21	2.16	2.12	2.09	2.06	2.02	1.97	1.92	1.90	1.87	1.85	1.82	1.76
16	3.05	2.67	2.46	2.33	2.24	2.18	2.13	2.09	2.06	2.03	1.99	1.94	1.89	1.87	1.84	1.81	1.78	1.72
17	3.03	2.64	2.44	2.31	2.22	2.15	2.10	2.06	2.03	2.00	1.96	1.91	1.86	1.84	1.81	1.78	1.75	1.69
18	3.01	2.62	2.42	2.29	2.20	2.13	2.08	2.04	2.00	1.98	1.93	1.89	1.84	1.81	1.78	1.75	1.72	1.66
19	2.99	2.61	2.40	2.27	2.18	2.11	2.06	2.02	1.98	1.96	1.91	1.86	1.81	1.79	1.76	1.73	1.70	1.63
20	2.97	2.59	2.38	2.25	2.16	2.09	2.04	2.00	1.96	1.94	1.89	1.84	1.79	1.77	1.74	1.71	1.68	1.61
21	2.96	2.57	2.36	2.23	2.14	2.08	2.02	1.98	1.95	1.92	1.87	1.83	1.78	1.75	1.72	1.69	1.66	1.59
22	2.95	2.56	2.35	2.22	2.13	2.06	2.01	1.97	1.93	1.90	1.86	1.81	1.76	1.73	1.70	1.67	1.64	1.57
23	2.94	2.55	2.34	2.21	2.11	2.05	1.99	1.95	1.92	1.89	1.84	1.80	1.74	1.72	1.69	1.66	1.62	1.55
24	2.93	2.54	2.33	2.19	2.10	2.04	1.98	1.94	1.91	1.88	1.83	1.78	1.73	1.70	1.67	1.64	1.61	1.53
25	2.92	2.53	2.32	2.18	2.09	2.02	1.97	1.93	1.89	1.87	1.82	1.77	1.72	1.69	1.66	1.63	1.59	1.52
26	2.91	2.52	2.31	2.17	2.08	2.01	1.96	1.92	1.88	1.86	1.81	1.76	1.71	1.68	1.65	1.61	1.58	1.50
27	2.90	2.51	2.30	2.17	2.07	2.00	1.95	1.91	1.87	1.85	1.80	1.75	1.70	1.67	1.64	1.60	1.57	1.49
28	2.89	2.50	2.29	2.16	2.06	2.00	1.94	1.90	1.87	1.84	1.79	1.74	1.69	1.66	1.63	1.59	1.56	1.48
29	2.89	2.50	2.28	2.15	2.06	1.99	1.93	1.89	1.86	1.83	1.78	1.73	1.68	1.65	1.62	1.58	1.55	1.47
30	2.88	2.49	2.28	2.14	2.05	1.98	1.93	1.88	1.85	1.82	1.77	1.72	1.67	1.64	1.61	1.57	1.54	1.46
40	2.84	2.44	2.23	2.09	2.00	1.93	1.87	1.83	1.79	1.76	1.71	1.66	1.61	1.57	1.54	1.51	1.47	1.38
60	2.79	2.39	2.18	2.04	1.95	1.87	1.82	1.77	1.74	1.71	1.66	1.61	1.54	1.51	1.48	1.44	1.40	1.29
∞	2.71	2.30	2.08	1.94	1.85	1.77	1.72	1.67	1.63	1.60	1.55	1.49	1.42	1.38	1.34	1.30	1.24	1.00

TABLE A.1 – Fractiles d'ordre $\alpha = 0.90 : f_{\nu_1, \nu_2; 0.90}$

$\nu_2 \backslash \nu_1$	1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	∞
1	161.4	199.5	215.7	224.6	230.2	234.0	236.8	238.9	240.5	241.9	243.9	245.9	248.0	249.1	250.1	251.1	252.2	395.4
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.41	19.43	19.45	19.45	19.46	19.47	19.48	19.50
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79	8.74	8.70	8.66	8.64	8.62	8.59	8.57	8.53
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.91	5.86	5.80	5.77	5.75	5.72	5.69	5.63
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74	4.68	4.62	4.56	4.53	4.50	4.46	4.43	4.37
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	4.00	3.94	3.87	3.84	3.81	3.77	3.74	3.67
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64	3.57	3.51	3.44	3.41	3.38	3.34	3.30	3.23
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35	3.28	3.22	3.15	3.12	3.08	3.04	3.01	2.93
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14	3.07	3.01	2.94	2.90	2.86	2.83	2.79	2.71
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98	2.91	2.85	2.77	2.74	2.70	2.66	2.62	2.54
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85	2.79	2.72	2.65	2.61	2.57	2.53	2.49	2.40
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75	2.69	2.62	2.54	2.51	2.47	2.43	2.38	2.30
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67	2.60	2.53	2.46	2.42	2.38	2.34	2.30	2.21
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.53	2.46	2.39	2.35	2.31	2.27	2.22	2.13
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.48	2.40	2.33	2.29	2.25	2.20	2.16	2.07
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.42	2.35	2.28	2.24	2.19	2.15	2.11	2.01
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.38	2.31	2.23	2.19	2.15	2.10	2.06	1.96
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.34	2.27	2.19	2.15	2.11	2.06	2.02	1.92
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.31	2.23	2.16	2.11	2.07	2.03	1.98	1.88
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.28	2.20	2.12	2.08	2.04	1.99	1.95	1.84
21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.25	2.18	2.10	2.05	2.01	1.96	1.92	1.81
22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30	2.23	2.15	2.07	2.03	1.98	1.94	1.89	1.78
23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27	2.20	2.13	2.05	2.01	1.96	1.91	1.86	1.76
24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25	2.18	2.11	2.03	1.98	1.94	1.89	1.84	1.73
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24	2.16	2.09	2.01	1.96	1.92	1.87	1.82	1.71
26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.15	2.07	1.99	1.95	1.90	1.85	1.80	1.69
27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20	2.13	2.06	1.97	1.93	1.88	1.84	1.79	1.67
28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19	2.12	2.04	1.96	1.91	1.87	1.82	1.77	1.65
29	4.18	3.33	2.93	2.70	2.55	2.43	2.35	2.28	2.22	2.18	2.10	2.03	1.94	1.90	1.85	1.81	1.75	1.64
30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	2.16	2.09	2.01	1.93	1.89	1.84	1.79	1.74	1.62
40	4.08	3.23	2.84	2.61	2.45	2.34	2.25	2.18	2.12	2.08	2.00	1.92	1.84	1.79	1.74	1.69	1.64	1.51
60	4.00	3.15	2.76	2.53	2.37	2.25	2.17	2.10	2.04	1.99	1.92	1.84	1.75	1.70	1.65	1.59	1.53	1.39
∞	3.84	3.00	2.60	2.37	2.21	2.10	2.01	1.94	1.88	1.83	1.75	1.67	1.57	1.52	1.46	1.39	1.32	1.00

TABLE A.2 – Fractiles d'ordre $\alpha = 0.95 : f_{\nu_1, \nu_2; 0.95}$

$\nu_2 \setminus \nu_1$	1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	∞
1	647.8	799.5	864.2	899.6	921.9	937.1	948.2	956.7	963.3	968.6	976.7	984.9	993.1	997.2	1001	1005	1009	1018
2	38.51	39.00	39.17	39.25	39.30	39.33	39.36	39.37	39.39	39.40	39.41	39.43	39.45	39.46	39.46	39.47	39.48	39.50
3	17.44	16.04	15.44	15.10	14.88	14.73	14.62	14.54	14.47	14.42	14.34	14.25	14.17	14.12	14.08	14.04	13.99	13.90
4	12.22	10.65	9.98	9.60	9.36	9.20	9.07	8.98	8.90	8.84	8.75	8.66	8.56	8.51	8.46	8.41	8.36	8.26
5	10.01	8.43	7.76	7.39	7.15	6.98	6.85	6.76	6.68	6.62	6.52	6.43	6.33	6.28	6.23	6.18	6.12	6.02
6	8.81	7.26	6.60	6.23	5.99	5.82	5.70	5.60	5.52	5.46	5.37	5.27	5.17	5.12	5.07	5.01	4.96	4.85
7	8.07	6.54	5.89	5.52	5.29	5.12	4.99	4.90	4.82	4.76	4.67	4.57	4.47	4.41	4.36	4.31	4.25	4.14
8	7.57	6.06	5.42	5.05	4.82	4.65	4.53	4.43	4.36	4.30	4.20	4.10	4.00	3.95	3.89	3.84	3.78	3.67
9	7.21	5.71	5.08	4.72	4.48	4.32	4.20	4.10	4.03	3.96	3.87	3.77	3.67	3.61	3.56	3.51	3.45	3.33
10	6.94	5.46	4.83	4.47	4.24	4.07	3.95	3.85	3.78	3.72	3.62	3.52	3.42	3.37	3.31	3.26	3.20	3.08
11	6.72	5.26	4.63	4.28	4.04	3.88	3.76	3.66	3.59	3.53	3.43	3.33	3.23	3.17	3.12	3.06	3.00	2.88
12	6.55	5.10	4.47	4.12	3.89	3.73	3.61	3.51	3.44	3.37	3.28	3.18	3.07	3.02	2.96	2.91	2.85	2.72
13	6.41	4.97	4.35	4.00	3.77	3.60	3.48	3.39	3.31	3.25	3.15	3.05	2.95	2.89	2.84	2.78	2.72	2.60
14	6.30	4.86	4.24	3.89	3.66	3.50	3.38	3.29	3.21	3.15	3.05	2.95	2.84	2.79	2.73	2.67	2.61	2.49
15	6.20	4.77	4.15	3.80	3.58	3.41	3.29	3.20	3.12	3.06	2.96	2.86	2.76	2.70	2.64	2.59	2.52	2.40
16	6.12	4.69	4.08	3.73	3.50	3.34	3.22	3.12	3.05	2.99	2.89	2.79	2.68	2.63	2.57	2.51	2.45	2.32
17	6.04	4.62	4.01	3.66	3.44	3.28	3.16	3.06	2.98	2.92	2.82	2.72	2.62	2.56	2.50	2.44	2.38	2.25
18	5.98	4.56	3.95	3.61	3.38	3.22	3.10	3.01	2.93	2.87	2.77	2.67	2.56	2.50	2.44	2.38	2.32	2.19
19	5.92	4.51	3.90	3.56	3.33	3.17	3.05	2.96	2.88	2.82	2.72	2.62	2.51	2.45	2.39	2.33	2.27	2.13
20	5.87	4.46	3.86	3.51	3.29	3.13	3.01	2.91	2.84	2.77	2.68	2.57	2.46	2.41	2.35	2.29	2.22	2.09
21	5.83	4.42	3.82	3.48	3.25	3.09	2.97	2.87	2.80	2.73	2.64	2.53	2.42	2.37	2.31	2.25	2.18	2.04
22	5.79	4.38	3.78	3.44	3.22	3.05	2.93	2.84	2.76	2.70	2.60	2.50	2.39	2.33	2.27	2.21	2.14	2.00
23	5.75	4.35	3.75	3.41	3.18	3.02	2.90	2.81	2.73	2.67	2.57	2.47	2.36	2.30	2.24	2.18	2.11	1.97
24	5.72	4.32	3.72	3.38	3.15	2.99	2.87	2.78	2.70	2.64	2.54	2.44	2.33	2.27	2.21	2.15	2.08	1.94
25	5.69	4.29	3.69	3.35	3.13	2.97	2.85	2.75	2.68	2.61	2.51	2.41	2.30	2.24	2.18	2.12	2.05	1.91
26	5.66	4.27	3.67	3.33	3.10	2.94	2.82	2.73	2.65	2.59	2.49	2.39	2.28	2.22	2.16	2.09	2.03	1.88
27	5.63	4.24	3.65	3.31	3.08	2.92	2.80	2.71	2.63	2.57	2.47	2.36	2.25	2.19	2.13	2.07	2.00	1.85
28	5.61	4.22	3.63	3.29	3.06	2.90	2.78	2.69	2.61	2.55	2.45	2.34	2.23	2.17	2.11	2.05	1.98	1.83
29	5.59	4.20	3.61	3.27	3.04	2.88	2.76	2.67	2.59	2.53	2.43	2.32	2.21	2.15	2.09	2.03	1.96	1.81
30	5.57	4.18	3.59	3.25	3.03	2.87	2.75	2.65	2.57	2.51	2.41	2.31	2.20	2.14	2.07	2.01	1.94	1.79
40	5.42	4.05	3.46	3.13	2.90	2.74	2.62	2.53	2.45	2.39	2.29	2.18	2.07	2.01	1.94	1.88	1.80	1.64
60	5.29	3.93	3.34	3.01	2.79	2.63	2.51	2.41	2.33	2.27	2.17	2.06	1.94	1.88	1.82	1.74	1.67	1.48
∞	5.02	3.69	3.12	2.79	2.57	2.41	2.29	2.19	2.11	2.05	1.94	1.83	1.71	1.64	1.57	1.48	1.39	1.00

TABLE A.3 – Fractiles d'ordre $\alpha = 0.975 : f_{\nu_1, \nu_2; 0.975}$

$\nu_2 \setminus \nu_1$	1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	∞
1	4052	4999.5	5403	5625	5764	5859	5928	5982	6022	6056	6106	6157	6209	6235	6261	6287	6313	6366
2	98.50	99.00	99.17	99.25	99.30	99.33	99.36	99.37	99.39	99.40	99.42	99.43	99.45	99.46	99.47	99.47	99.48	99.59
3	34.12	30.82	29.46	28.71	28.24	27.91	27.67	27.49	27.35	27.23	27.05	26.87	26.69	26.60	26.50	26.41	26.32	26.12
4	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.66	14.55	14.37	14.20	14.02	13.93	13.84	13.75	13.65	13.46
5	16.26	13.27	12.06	11.39	10.97	10.67	10.46	10.29	10.16	10.05	9.89	9.72	9.55	9.47	9.38	9.29	9.20	9.02
6	13.75	10.92	9.78	9.15	8.75	8.47	8.26	8.10	7.98	7.87	7.72	7.56	7.40	7.31	7.23	7.14	7.06	6.88
7	12.25	9.55	8.45	7.85	7.46	7.19	6.99	6.84	6.72	6.62	6.47	6.31	6.16	6.07	5.99	5.91	5.82	5.65
8	11.26	8.65	7.59	7.01	6.63	6.37	6.18	6.03	5.91	5.81	5.67	5.52	5.36	5.28	5.20	5.12	5.03	4.86
9	10.56	8.02	6.99	6.42	6.06	5.80	5.61	5.47	5.35	5.26	5.11	4.96	4.81	4.73	4.65	4.57	4.48	4.31
10	10.04	7.56	6.55	5.99	5.64	5.39	5.20	5.06	4.94	4.85	4.71	4.56	4.41	4.33	4.25	4.17	4.08	3.91
11	9.65	7.21	6.22	5.67	5.32	5.07	4.89	4.74	4.63	4.54	4.40	4.25	4.10	4.02	3.94	3.86	3.78	3.60
12	9.33	6.93	5.95	5.41	5.06	4.82	4.64	4.50	4.39	4.30	4.16	4.01	3.86	3.78	3.70	3.62	3.54	3.36
13	9.07	6.70	5.74	5.21	4.86	4.62	4.44	4.30	4.19	4.10	3.96	3.82	3.66	3.59	3.51	3.43	3.34	3.17
14	8.86	6.51	5.56	5.04	4.69	4.46	4.28	4.14	4.03	3.94	3.80	3.66	3.51	3.43	3.35	3.27	3.18	3.00
15	8.68	6.36	5.42	4.89	4.56	4.32	4.14	4.00	3.89	3.80	3.67	3.52	3.37	3.29	3.21	3.13	3.05	2.87
16	8.53	6.23	5.29	4.77	4.44	4.20	4.03	3.89	3.78	3.69	3.55	3.41	3.26	3.18	3.10	3.02	2.93	2.75
17	8.40	6.11	5.18	4.67	4.34	4.10	3.93	3.79	3.68	3.59	3.46	3.31	3.16	3.08	3.00	2.92	2.83	2.65
18	8.29	6.01	5.09	4.58	4.25	4.01	3.84	3.71	3.60	3.51	3.37	3.23	3.08	3.00	2.92	2.84	2.75	2.57
19	8.18	5.93	5.01	4.50	4.17	3.94	3.77	3.63	3.52	3.43	3.30	3.15	3.00	2.92	2.84	2.76	2.67	2.49
20	8.10	5.85	4.94	4.43	4.10	3.87	3.70	3.56	3.46	3.37	3.23	3.09	2.94	2.86	2.78	2.69	2.61	2.42
21	8.02	5.78	4.87	4.37	4.04	3.81	3.64	3.51	3.40	3.31	3.17	3.03	2.88	2.80	2.72	2.64	2.55	2.36
22	7.95	5.72	4.82	4.31	3.99	3.76	3.59	3.45	3.35	3.26	3.12	2.98	2.83	2.75	2.67	2.58	2.50	2.31
23	7.88	5.66	4.76	4.26	3.94	3.71	3.54	3.41	3.30	3.21	3.07	2.93	2.78	2.70	2.62	2.54	2.45	2.26
24	7.82	5.61	4.72	4.22	3.90	3.67	3.50	3.36	3.26	3.17	3.03	2.89	2.74	2.66	2.58	2.49	2.40	2.21
25	7.77	5.57	4.68	4.18	3.85	3.63	3.46	3.32	3.22	3.13	2.99	2.85	2.70	2.62	2.54	2.45	2.36	2.17
26	7.72	5.53	4.64	4.14	3.82	3.59	3.42	3.29	3.18	3.09	2.96	2.81	2.66	2.58	2.50	2.42	2.33	2.13
27	7.68	5.49	4.60	4.11	3.78	3.56	3.39	3.26	3.15	3.06	2.93	2.78	2.63	2.55	2.47	2.38	2.29	2.10
28	7.64	5.45	4.57	4.07	3.75	3.53	3.36	3.23	3.12	3.03	2.90	2.75	2.60	2.52	2.44	2.35	2.26	2.06
29	7.60	5.42	4.54	4.04	3.73	3.50	3.33	3.20	3.09	3.00	2.87	2.73	2.57	2.49	2.41	2.33	2.23	2.03
30	7.56	5.39	4.51	4.02	3.70	3.47	3.30	3.17	3.07	2.98	2.84	2.70	2.55	2.47	2.39	2.30	2.21	2.01
40	7.31	5.18	4.31	3.83	3.51	3.29	3.12	2.99	2.89	2.80	2.66	2.52	2.37	2.29	2.20	2.11	2.02	1.80
60	7.08	4.98	4.13	3.65	3.34	3.12	2.95	2.82	2.72	2.63	2.50	2.35	2.20	2.12	2.03	1.94	1.84	1.60
∞	6.63	4.61	3.78	3.32	3.02	2.80	2.64	2.51	2.41	2.32	2.18	2.04	1.88	1.79	1.70	1.59	1.47	1.00

TABLE A.4 – Fractiles d'ordre $\alpha = 0.99$: $f_{\nu_1, \nu_2; 0.99}$

A.6 Distribution de Kolmogorov-Smirnov

La table donne les fractiles d'ordre α de la loi de Kolmogorov-Smirnov pour une taille n d'échantillon, c'est-à-dire $\mathbb{P}(D_n \geq d_{n,1-\alpha}) = \alpha$ où $D_n = \sup |F_n - F_0|$.

$n \setminus 1-\alpha$.80	.85	.90	.95	.99
1	.900	.925	.950	.975	.995
2	.684	.726	.776	.842	.929
3	.565	.597	.642	.708	.829
4	.494	.525	.564	.624	.734
5	.446	.474	.510	.563	.669
6	.410	.436	.470	.521	.618
7	.381	.405	.438	.486	.577
8	.358	.381	.411	.457	.543
9	.339	.360	.388	.432	.514
10	.322	.342	.368	.409	.486
11	.307	.326	.352	.391	.468
12	.295	.313	.338	.375	.450
13	.254	.302	.325	.361	.433
14	.274	.292	.314	.349	.418
15	.266	.283	.304	.338	.404
16	.258	.274	.295	.328	.391
17	.250	.266	.286	.318	.380
18	.244	.259	.278	.309	.370
19	.237	.252	.272	.301	.361
20	.231	.246	.264	.294	.352
25	.210	.220	.240	.264	.320
30	.190	.200	.220	.242	.290
35	.180	.190	.210	.230	.270
40				.210	.250
50				.190	.23
60				.170	.21
70				.160	.19
80				.150	.18
90				.140	
100				.140	
∞	$\frac{1.07}{\sqrt{n}}$	$\frac{1.14}{\sqrt{n}}$	$\frac{1.22}{\sqrt{n}}$	$\frac{1.36}{\sqrt{n}}$	$\frac{1.63}{\sqrt{n}}$

TABLE A.5 – Fractiles $d_{n,1-\alpha}$ de D_n

Formulaire de statistique

B.1 Statistiques pour une population normale de taille n

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1), \quad \sigma \text{ connue}; \quad \frac{\bar{X} - \mu}{S^*/\sqrt{n}} \sim \mathcal{T}_{n-1}, \quad \sigma \text{ inconnue.}$$

$$\frac{\sum_i (X_i - \mu)^2}{\sigma^2} \sim \chi_n^2, \quad \mu \text{ connue}; \quad \frac{\sum_i (X_i - \bar{X})^2}{\sigma^2} \sim \chi_{n-1}^2, \quad \mu \text{ inconnue.}$$

B.2 Statistiques pour 2 populations normales de tailles n, m

$$\frac{S_X^{*2}/\sigma_X^2}{S_Y^{*2}/\sigma_Y^2} \sim \mathcal{F}_{n-1; m-1}; \quad \frac{\bar{X} - \bar{Y} - (\mu_x - \mu_y)}{S^* \sqrt{\frac{1}{n} + \frac{1}{m}}} \sim \mathcal{T}_{n+m-2},$$

où

$$S^{*2} = \frac{(n-1)S_X^{*2} + (m-1)S_Y^{*2}}{n+m-2}.$$

B.3 Analyse de la variance à 1 facteur

B.3.1 Statistiques

	effectif	somme des obs.	somme des carrés des obs.
groupe q	n_q	$SX_q = \sum_{i=1}^{n_q} X_{qi}$	$SX2_q = \sum_{i=1}^{n_q} X_{qi}^2$
total	$n = \sum_{q=1}^Q n_q$	$SX = \sum_{q,i} X_{qi}$	$SX2 = \sum_{q,i} X_{qi}^2$

B.3.2 Synthèse

source	ddl	SC	CM	R
groupes	$Q - 1$	$SCF = SCT - SCR$	$CMF = \frac{SCF}{Q-1}$	$\frac{CMF}{CMR}$
résidus	$n - Q$	$SCR = \sum_q SX2_q - \frac{1}{n_q} (SX_q)^2$	$CMR = \frac{SCR}{n-Q}$	
total	$n - 1$	$SCT = SX2 - \frac{1}{n} (SX)^2$		

$$R \sim \mathcal{F}_{Q-1; n-Q}$$

B.3.3 Contrastes

$$\frac{X_{q\bullet} - X_{l\bullet} - C_{ql}}{\sqrt{CMR(\frac{1}{n_q} + \frac{1}{n_l})}} \sim \mathcal{T}_{n-Q}.$$

B.4 Analyse de la variance à 2 facteurs

B.4.1 Statistiques

$$SCA = \sum_{q=1}^a n_{q+} X_{q\bullet}^2 - n X_{\bullet\bullet}^2$$

$$SCB = \sum_{\ell=1}^b n_{+\ell} X_{\bullet\ell}^2 - n X_{\bullet\bullet}^2$$

$$SCAB = \sum_{q\ell} n_{q\ell} (X_{q\ell} - X_{q\bullet} - X_{\bullet\ell} + X_{\bullet\bullet})^2$$

B.4.2 Modèle général

source	ddl	SC	CM	F
Facteur A	$a - 1$	SCA	$\frac{SCA}{a-1}$	$R_A = \frac{(n-ab)SCA}{(a-1)SCR}$
Facteur B	$b - 1$	SCB	$\frac{SCB}{b-1}$	$R_B = \frac{(n-ab)SCB}{(b-1)SCR}$
Interaction	$(a - 1)(b - 1)$	$SCAB$	$\frac{SCAB}{(a-1)(b-1)}$	$R_{AB} = \frac{(n-ab)SCAB}{(a-1)(b-1)SCR}$
Résiduelle	$n - ab$	SCR	$\frac{SCR}{(n-ab)}$	
Total	$n - 1$	SCT		

$$R_{AB} \sim \mathcal{F}_{(a-1)(b-1), (n-ab)}$$

B.4.3 Modèle additif

source	ddl	SC	CM	F
Facteur A	$a - 1$	SCA	$\frac{SCA}{a-1}$	$R_A = \frac{(n-a-b+1)SCA}{(a-1)SCR}$
Facteur B	$b - 1$	SCB	$\frac{SCB}{b-1}$	$R_B = \frac{(n-a-b+1)SCB}{(b-1)SCR}$
Résiduelle	$n - a - b + 1$	SCR	$\frac{SCR}{(n-a-b+1)}$	
Total	$n - 1$	SCT		

On obtient la somme des carrés résiduelle via $SCR = SCT - SCA - SCB$.

B.4.4 Modèles emboîtés

Le test

$$\begin{cases} H_0 : \text{modèle } \omega \text{ est correct} \\ H_1 : \text{modèle } \Omega \text{ est correct.} \end{cases}$$

admet pour région critique

$$\Gamma_\alpha = \left\{ \frac{(SCR(\omega) - SCR(\Omega))(n - ab)}{SCR(ddl_\Omega - ddl_\omega)} > f_{(ddl_\Omega - ddl_\omega, n - ab; 1 - \alpha)} \right\}.$$

Régression linéaire simple

B.4.5 Modèle

$$Y = \beta_0 + \beta_1 x + \varepsilon \text{ avec } \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

B.4.6 Statistiques

$$\begin{cases} \hat{\beta}_0 = \bar{Y} - \frac{S_{xY}}{S_x^2} \bar{x}, \\ \hat{\beta}_1 = \frac{S_{xY}}{S_x^2}, \end{cases}$$

avec

$$S_{xY} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y}), \quad S_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

$$\hat{\beta}_1 \sim \mathcal{N}\left(\beta_1, \frac{\sigma^2}{nS_x^2}\right), \quad \hat{\beta}_0 \sim \mathcal{N}\left(\beta_0, \frac{\sigma^2}{n}\left(1 + \frac{\bar{x}^2}{S_x^2}\right)\right), \quad \text{cov}(\hat{\beta}_0, \hat{\beta}_1) = -\frac{\bar{x}\sigma^2}{nS_x^2}.$$

B.4.7 Synthèse

source	ddl	SC	CM	F
Modèle	1	<i>SCM</i>	<i>SCM</i>	$F = \frac{(n-2)SCM}{SCR}$
Résiduelle	$n - 2$	<i>SCR</i>	$\frac{SCR}{(n-2)}$	
Total	$n - 1$	<i>SCT</i>		

$$R^2 = \frac{SCM}{SCT}$$

B.4.8 Intervalle de confiance sur $\mathbb{E}(Y_o)$

$$\frac{\hat{Y}_o - \mathbb{E}(Y_o)}{\sqrt{CMR\left(\frac{1}{n} + \frac{(\bar{x} - x_o)^2}{nS_x^2}\right)}} \sim \mathcal{T}_{n-2}.$$

B.4.9 Prédiction

$$\frac{\hat{Y}_o - Y_o}{\sqrt{CMR\left(1 + \frac{1}{n} + \frac{(\bar{x} - x_o)^2}{nS_x^2}\right)}} \sim \mathcal{T}_{n-2}.$$